Selecting Classification Features for Detection of Mass Emergency Events on Social Media

V. Pekar[1], J. Binner[1], H. Najafi[2], and C. Hale[3]
[1]Business School, University of Birmingham, Birmingham, United Kingdom
[2]Computer Science and Information Systems, University of Wisconsin, River Falls, WI, USA
[3]Electronic Systems Lab, Georgia Tech Research Institute, Dayton, OH, USA

Abstract *The paper addresses the problem of detecting eyewitness reports of mass emergencies on Twitter. This is the first work to conduct a large-scale comparative evaluation of classification features extracted from Twitter posts, using different learning algorithms and datasets representing a broad range of mass emergencies including both natural and technological disasters. We investigate the relative importance of different feature types as well as on the effect of several feature selection methods applied to this problem. Because the task of detecting mass emergencies is characterized by high heterogeneity of the data, our primary focus is on identifying those features that are capable of separating mass emergency reports from other messages, irrespective of the type of the disaster*.

Keywords: text classification, feature selection, social media analysis, disaster management

1 Introduction

Social media data offer a promising possibility to deal with mass emergencies. The present-day ubiquity of mobile devices has meant that during a crisis such as a flood, earthquake or a terrorist attack, social media becomes a primary source of information, publishing eyewitness reports on the events in real-time. This gives an opportunity for emergency services to detect crises at early stages, monitor their development and tackle their consequences more effectively.

The potential of social media analysis for mass emergency management has attracted many Data Mining researchers over the past several years. The problem of detecting eyewitness accounts of emergency events in social media has been primarily approached with text classification methods based on machine learning. Limiting the problem to a narrow domain such as earthquakes or tornados has been shown to produce high classification accuracy (e.g., [3, 4, 8, 11, 12, 22]).

However, mass emergency events differ a lot and a classification method that would cover a wide range of possible disasters would be much more practical. This paper is concerned with the broader task of recognizing emergencies unspecified for a particular type, which could include both natural disasters such as earthquakes, floods and storms, as well as man-made ones such as explosions, collisions and shootings. This is a non-trivial classification problem. Firstly, messages relating to a crisis event include not only actual eyewitness accounts but also those that have to do with official announcements, offers of help, sympathy, criticism, and so on. Olteanu et al. [12] report that of all messages judged to be relevant to one of twenty-six mass emergencies, eyewitness accounts comprise only around 8%. The challenge is therefore in identifying specifically eyewitness reports among messages that talk about largely the same event; this is also a classification problem with a big bias towards the negative class. Secondly, because the automatic classifier is expected to operate on a broad variety of event types, each characterized by its own vocabulary. The data is thus not homogeneous: data instances come from related, but different distributions, and in real-world use cases training data is likely to be insufficiently representative of test data on which the classifier is evaluated.

To address these challenges, we study classification features that can be extracted from Twitter messages, beyond the traditional text-based features, that would be suited specifically to the task at hand. Until now, previous papers on detecting emergency-related messages used their own set of features; a few studies examined their contribution to classifier accuracy, but only within a specific application, often limited to one learning algorithm and one emergency event. In this paper we describe a comparative evaluation of a broad set of features that includes those that were used in previous work as well as those introduced for the first time, conducting experiments on data from 26 different emergency events. We report on features that are robust against data heterogeneity and help achieve better classification accuracy when the classifier is evaluated on data from an emergency event that is different from the events exemplified in the training data.

2 Related Work

There is a considerable body of work on detection of new events in a stream of messages, where the type of the event of interest is not known in advance, and some of these approaches were applied to detecting mass emergency events.

Such methods primarily rely on detecting "bursty" keywords [10], i.e. keywords whose frequency increases sharply within a short time window, or bursty message clusters [16]. However, bursty keywords are known to be related not only to new events, but also recurring events and even non-events. To separate them, Becker et al. [2] used a domain-independent text classifier, before applying keyword burstiness techniques.

Domain-specific methods generally have a greater accuracy than domain-independent ones, and previous work specifically on emergency event detection was concerned with developing domain text classifiers based on machine learning and operating on features extracted from the entire message. Most of this work dealt with specific types of crises such as earthquakes [3, 21, 22, 23], bushfires [14], tornados [4, 8], and landslides [11]. Only a few studies developed classifiers that would be applied to more than one type of disasters: Verma et al. [20] evaluated their method on three different types of crises, while Ashktorab et al. [1] on five.

Classification features typically include unigrams (e.g., [1, 15]), bigrams [20, 22], message length [11, 15], part-of-speech tags [4, 19], VerbNet categories [4], the proportion of words that are present in a pre-defined vocabulary [11], whether place names are present [11], hashtags [4, 22], if the message is a retweet [4, 22] or a reply [2]. Verma at al. [20] looked at the contribution of three other kinds of features: if the language of the message objective or subjective, if the register is formal or not, if the text is a first-person report or not.

Any direct comparison between previous approaches is difficult, because they used different experimental datasets, different classification algorithms, and the classification tasks were somewhat different. For example, Imran et al. [5] classified messages into "informative" and "non-informative", Ashktorab et al. [1] into those that report damage and those that do not, Verma et al. [20] into those that are related to situational awareness and those that are not.

3 Classification features

In our evaluation we include the following types of features (examples are shown in parentheses):

Lexical:

`Unigrams`: whitespace-separated word tokens (nominal: *please, help, fire*).

`Bigrams`: token sequences with the length of two (nominal: *was_scary, we_complained*).

`NumberOfUnigrams`: the length of the messages, measured in unigrams. Sakaki et al. [14] found that it was a useful class predictor, as in their data eye-witness accounts tended to be short messages (continuous).

Grammatical:

`Verbs&Nouns`: only word tokens that are tagged as verbs and nouns. The intuition behind these features is that events and their participants are usually described with verbs and nouns, and thus events can be more accurately classified by focusing on verbs and nouns found in the message (nominal: *construction, floor, stuck*).

`PartOfSpeechTags`: separate features are created from part-of-speech (PoS) categories, as assigned by a PoS tagger, the reasoning being that the greater incidence of specific parts of speech (e.g., verbs and nouns) may be more indicative of an eye-witness report (nominal: *NNS, JJ, VBD*).

Semantic:

`VerbNetCategories`: VerbNet [6] is a lexical resource encoding English verbs and different semantic information on them, including their semantic categories. Following Imran et al. [4], for each verb found in a tweet, we add a feature corresponding to its VerbNet category in order to generalize the meaning of specific verbs (nominal: *complain-37.8, get-13.5.1*).

`EMTCategories`: Emergency Management Terms [19] is a lexicon containing around 7,000 words and expressions semi-automatically extracted from Twitter messages on different public emergencies. Each item in the lexicon is associated with a semantic category such as "Caution and Advice", "Injured People", "Infrastructure damage". We detect EM terms in the tweets and use their category labels as features (nominal: *T04, T07, O02*).

`NamedEntities`: We map all named entities, as detected and tagged by the PoS tagger, to a category label, and use it as a feature, instead of actual word tokens (nominal).

Stylistic:

`Sentiment`: We process each tweet with a domain-independent sentiment analysis system [13] and create a feature indicating whether the tweet is neutral in terms of sentiment or not; the system detects emoticons and uses them to determine the sentiment of the message (Boolean).

`Personal`: Following Verma et al. [20], we create a feature indicating if the message contains first-person pronouns ("I", "we", "me", "us") or not, expecting that eyewitness accounts of emergencies will be written from a first-person perspective (Boolean).

`All caps`: We create a feature indicating if the tweet contains all-caps words or not, as words spelled all in capital letters are meant to represent shouting, i.e. used when the author wants to attract special attention to the tweet (Boolean).

Twitter metadata:

`Hashtags`: A hashtag is a word or concatenated phrase preceded by the hash symbol, which are used by authors of messages to group tweets on the same topic and indicate important keywords; we create one extra feature for each hashtag found in a tweet (nominal: *#sandy, #haze*).

`ContainsHashtags`: whether or not the tweet contains any hashtags (Boolean).

`Mentions`: A mention is the name of a Twitter account that is included into the message in order to attract that user's attention to the tweet. We hypothesize that in case crises are reported, the tweet would mention the same set of Twitter

accounts (e.g., news agencies, police, or government bodies). We create one feature for each mention found in the tweet (nominal: *@newscaster, @News1130radio*).

`ContainsMentions`: whether or not the message mentions one or several Twitter accounts. Becker et al [2] found that presence of mentions correlates with reports of emergency events (Boolean).

`RetweetCount`: the number of times the message has been retweeted. We anticipate that eyewitness accounts are likely to attract more interest than other tweets and thus would be retweeted more (continuous).

`Reply`: whether the message is a reply to a different message. In accordance with Becker et al.'s findings [2], we expect that eyewitness accounts will not be replies to previous messages (Boolean).

`ContainsURL`: whether the tweet contains a URL. We expect that eyewitness accounts will tend not to mention any previously published information such as external URLs (Boolean).

Prior to training and classification, all features are converted to the continuous values.

4 Feature selection

Feature selection is a common step in machine learning scenarios, and in particular in text classification, where the number of features is usually very large. It is performed in order to eliminate noisy features, minimize overfitting of the classifier to the training data and to improve its efficiency. In supervised settings, i.e., when class membership of instances is known, the *filtering* approach to feature selection is commonly followed. For an overview of feature selection methods used in text classification, see [17].

In the context of tweet classification the filtering approach can be formalized as follows. Let us assume that each tweet $t \in T$ of the training set is represented as a feature vector, consisting of features $f \in F$, and that each t is assigned a class label $c \in C$. For each f, from its distribution across C, a certain function computes its informativeness score $s(f,c)$, specific to each class. From class-specific scores, one can compute its global score by, e.g. averaging local scores of f across classes. The features are then sorted by their informativeness and k top features are selected to represent instances, with k set experimentally. After non-informative features have been removed from the training data, a classifier is learned and evaluated on the test data.

A key decision for feature selection is to choose a function computing $s(f,c)$. Such functions aim to capture the intuition that the best features for a class are the ones that best discriminate between its positive and negative examples. They determine $s(f,c)$ from the distribution of f between c and non-c, attributing greater weights to those f that correlate with c or non-c the most. In the present study we include three such functions widely used in text categorization.

Chi-square. The chi-square (CHI) statistic measures the lack of independence between f and c, and can be used directly as the informativeness score. The chi-square is calculated between the observed frequency of co-occurrence of f and c $fr(f, c)$ and their expected co-occurrence $fr'(f, c)$. First the latter is obtained assuming the f and c co-occur randomly:

$$fr'(f,c) = \frac{fr(f) \cdot fr(c)}{\sum_{m \in F} \sum_{n \in C} fr(m,n)}$$

The chi-square statistic is then calculated as:

$$\chi^2(f,c) = \sum_{m \in \{f, \bar{f}\}} \sum_{n \in \{c, \bar{c}\}} \frac{(fr(m,n) - fr'(m,n))^2}{fr'(m,n)}$$

Information Gain. IG measures the number of bits of information obtained about presence and absence of c by knowing the presence or absence of f. It is calculated as follows:

$$IG(f,c) = \sum_{m \in \{f, \bar{f}\}} \sum_{n \in \{c, \bar{c}\}} p(m,n) \log \frac{p(m,n)}{p(m)p(n)}$$

Information Gain Ratio. IGR is a normalized version of IG, meant to overcome the disadvantage of IG that it grows not only with the increase of dependence between f and c, but also with the increase of the entropy of f. IGR removes this factor by normalizing IG by the entropy of c:

$$GR(f,c) = \frac{IG(f,c)}{-\sum_{n \in \{c, \bar{c}\}} p(n) \log p(n)}$$

5 Experimental setup

5.1 Data

For experimental evaluation we use the labeled part of the CrisisLexT26 dataset [12], which includes tweets on 26 mass emergencies that occurred in 2012 and 2013. The types of emergencies are very diverse and range from terrorist attacks and train derailment to floods and hurricanes. Some examples are Colorado wildfires in 2012, Venezuela refinery explosion in 2012, Boston bombings in 2013.

There are 24,589 labeled tweets in the dataset in total, with 2,193 of them labeled as originating from an eyewitness. The classification task in our experiments consisted of predicting whether a given tweet was an eyewitness report or not.

5.2 Preprocessing

We apply the following preprocessing steps to the data:

Additional metadata. The CrisisLexT26 data contains the Twitter id of the message, its raw content, and its timestamp.

Via Twitter Search API we retrieve additional metadata fields: retweet count, reply, hashtags.

Deduplication. Duplicate tweets were removed by measuring similarity in each pair of tweets using cosine and removing one tweet in pairs where the cosine was higher than 0.99.

Tokenization and part-of-speech tagging. Before processing the text of the message with a PoS tagger, the text was normalized: mentions (e.g., @someone) and URLs removed; sequences of hashtags at the start and end of the message removed; hashtags appearing in the middle of the text were kept, but the hash symbol removed from the hashtags; long non-alphanumeric symbol sequences, which tend to be emoticons, were removed; word tokens consisting of digits were replaced with a unique tag. The normalized text was tokenized and tagged with the PoS tagger in the Pattern library [19].

Sentiment analysis. The original text of the tweet was processed with the sentiment analysis system [13]. The system was used in the SemEval ABSA challenge, where it achieved an F-measure of 0.67 and 0.75 on the two evaluation datasets within the sentiment analysis subtask. The system assigned to an input text a sentiment score between -1.0 (negative) and 1.0 (positive); the score was converted to a Boolean value indicating if the tweet is neutral in terms of sentiment (the score was equal 0) or not.

Stopword removal. The usual stoplist was used to remove stopwords.

5.3 Evaluation Metrics

The accuracy of classification is measured in terms of the traditional measures of precision, recall and F1 measure. Because the data is biased towards the negative class, the evaluation metrics averaged over both classes may be misleading, so we report them only for the positive class, i.e. the eyewitness report class.

6 Results and Discussion

6.1 Impact of Data Heterogeneity

In the first experiment, we examined the extent to which data heterogeneity present in the CrisisLexT2 dataset affects classification accuracy. To that end, we evaluated the classifiers in two scenarios. In the first ("Scenario 1"), the entire dataset was randomly split into a train and a test set, in proportion 1 to 9. This ensured, with a large likelihood, that data on the same crisis will be present in both training and test data, and the feature distribution in the test data will be similar to the one in the train data.

The second scenario ("Scenario 2") was meant to better reflect real-world use cases: the train-test split was done in such a way so that the test data contained tweets only on those crises that were not included into the train data, i.e., simulating the conditions when a crisis needs to be detected before any manually labelled data relating to it are available. Specifically, data on 23 crises were used for training and data on 3 remaining crises were used for testing.

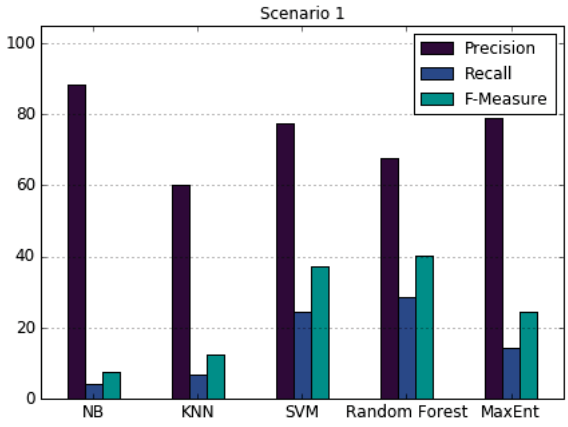

Figure 1. Classifier performance on the full set of features, random train-test split ("Scenario 1").

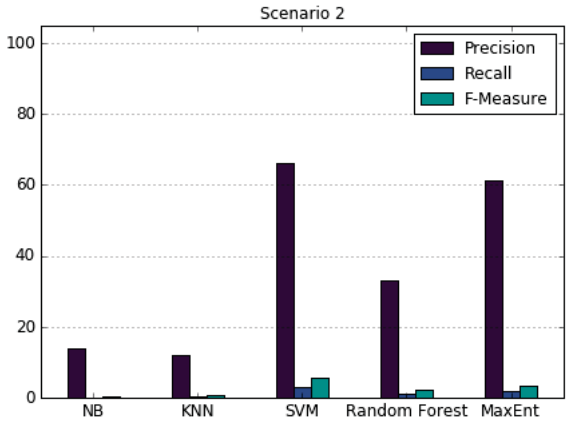

Figure 2. Classifier performance on the full set of features ("Scenario 2").

Training on all the features described in Section 3, we compared the performance of five classifiers – Naive Bayes, k Nearest Neighbors (kNN, k=5), Random Forest, Maximum Entropy (MaxEnt, a.k.a. Logistic regression) and linear Support Vector Machines (SVM), under these two scenarios. The results are shown in Figures 1 and 2

The results show that Scenario 2 is indeed a much harder evaluation task: both precision and recall rates for all the five classifiers drop; the drop is especially big for recall (e.g., for SVM it falls 0.24 from to 0.01). This suggests that, as anticipated, there is more data heterogeneity between different crises than within them. To confirm this, we measured the difference between distributions of features in each train-test split using Jensen-Shannon divergence, a variant of Kullback-Leibler divergence [9]; feature probabilities are obtained via Maximum Likelihood Estimation. We find that the average JS divergence in Scenario 1 is 0.01, while in Scenario 2 it is much higher, at 0.07, the difference is significant based on an independent samples t-test ($p < 0.001$).

In our subsequent experiments, we used the SVM and MaxEnt classifiers, which fared better than the other three.

6.2 Feature types

To measure the relative utility of each type of features, we ran experiments where each feature type was removed from the full set of features and the change in classifier performance was noted. The results for Scenario 1 are shown in Tables 1 (SVM) and 2 (MaxEnt), for Scenario 2 – in Tables 3 (SVM) and 4 (MaxEnt), the tables show the percent changes in F-measure, precision and recall resulting from removing one feature type.

	F-measure	Precision	Recall
Bigrams	-3.39	0.70	-2.91
Hashtags	-3.10	0.16	-2.53
Mentions	-1.51	-1.47	-1.18
ContainsMentions	-0.83	-0.08	-0.63
EMCategories	-0.78	-0.75	-0.58
Sentiment	-0.39	0.70	-0.39
AllCaps	-0.35	-0.71	-0.22
Personal	-0.34	0.05	-0.29
Reply	-0.23	0.54	-0.21
PosTags	-0.20	-0.98	-0.08
ContainsUrl	-0.16	-0.11	-0.12
Verbnet	0.03	-0.15	0.03
ContainsHashtags	0.22	0.10	0.21
VerbsAndNouns	0.61	-1.19	0.70
NumberOfUnigrams	0.94	-0.80	0.99
NamedEntities	3.10	1.14	2.67
RetweetCount	9.05	-0.22	8.51

Table 1. The effects of removing one feature from the feature set, Scenario 1, SVM.

	F-measure	Precision	Recall
PosTags	-2.66	-0.77	-1.79
Hashtags	-2.42	1.01	-1.62
EMCategories	-1.19	1.29	-0.84
ContainsHashtags	-1.14	-0.16	-0.76
Personal	-0.82	-0.72	-0.54
AllCaps	-0.32	-0.74	-0.16
Mentions	-0.28	-0.55	-0.15
Sentiment	-0.17	-1.33	-0.07
Reply	-0.14	0.0	-0.05
ContainsMention	0.04	0.68	0.02
ContainsUrl	0.04	-1.78	0.13
Verbnet	0.07	-0.64	0.08
VerbsAndNouns	0.44	-0.42	0.32
Bigrams	1.84	-4.92	1.54
NumberOfUnigrams	2.50	-1.80	1.87
NamedEntities	3.61	1.40	2.55
RetweetCount	9.50	2.57	7.04

Table 2. The effects of removing one feature from the feature set, Scenario 1, MaxEnt.

For Scenario 1 results, we see that the changes are not very significant, except for `Hashtags`, which contribute a lot to the recall of the classifiers (up to 5 points), `Bigrams`, which help precision for Maxent and recall for SVM, and `RetweetCount` and `NamedEntities`, whose removal leads to improvements in all the three metrics, by up to 9 points. Some features increase precision at the cost of recall (`NumberOfUnigrams`), while others, on the contrary, improve recall at the cost of precision (`HashTags`, `EMCategories`).

For Scenario 2, the changes in F-measure are not high, but differences between specific features in terms of precision and recall are much more noticeable than for Scenario 1. For both classifiers, `Bigrams` are important for precision, the changes are 11.7 points for SVM and 8.2 for MaxEnt, while `EMCategories` and `Personal` help to improve recall. The use of `PosTags` improves all the evaluation metrics for both classifiers. `RetweetCount`, `VerbNet`, `HashTags` and `VerbsAndNouns` produce an adverse effect on all the three metrics, also for both SVM and MaxEnt. A somewhat unexpected observation is that `PosTags` are positively influencing all the metrics, for all classifiers and scenarios. The changes for other features are less consistent between the classifiers.

	F-measure	Precision	Recall
Bigrams	-1.97	-11.70	-1.08
Personal	-0.97	-5.20	-0.55
NamedEntities	-0.26	-2.48	-0.15
ContainsHashtags	-0.25	6.82	-0.17
PosTags	-0.22	-3.44	-0.12
EMCategories	-0.20	2.20	-0.13
Sentiment	-0.18	-2.44	-0.11
ContainsMention	-0.15	-0.05	-0.09
ContainsUrl	-0.02	1.77	-0.04
Reply	0.0	0.0	0.0
Mentions	0.03	3.71	0.01
AllCaps	0.03	-0.85	0.01
Verbnet	0.22	0.44	0.11
Hashtags	0.53	9.39	0.26
RetweetCount	1.25	-3.49	0.75
NumberOfUnigrams	1.3	-5.16	0.76
VerbsAndNouns	1.58	6.46	0.85

Table 3. The effects of removing one feature from the feature set, Scenario 2, SVM.

More generally, it seems that lexical features such as `Bigrams` help to achieve greater precision, while Semantic (e.g., `EMCategories`), Stylistic (e.g., `Personal`) and Twitter-related (e.g., `ContainsHashtags`) ones – greater recall. These characteristics of the features become more prominent in Scenario 2.

	F-measure	Precision	Recall
EMCategories	-1.13	2.24	-0.63
PosTags	-0.97	-38.77	-0.51
ContainsHashtags	-0.68	-2.0	-0.38
Personal	-0.68	5.52	-0.39
ContainsUrl	-0.34	-4.71	-0.19
NamedEntities	-0.31	2.31	-0.19
Sentiment	0.0	0.0	0.0
Mentions	0.0	0.0	0.0
AllCaps	0.14	1.17	0.07
Reply	0.15	1.85	0.07
VerbsAndNouns	0.37	5.27	0.18
ContainsMention	0.40	2.22	0.22
Bigrams	0.62	-8.24	0.36
Verbnet	0.76	2.46	0.41
Hashtags	0.93	2.77	0.50
RetweetCount	0.98	-8.69	0.56
NumberOfUnigrams	1.55	2.45	0.85

Table 4. The effects of removing one feature from the feature set, Scenario 2, MaxEnt.

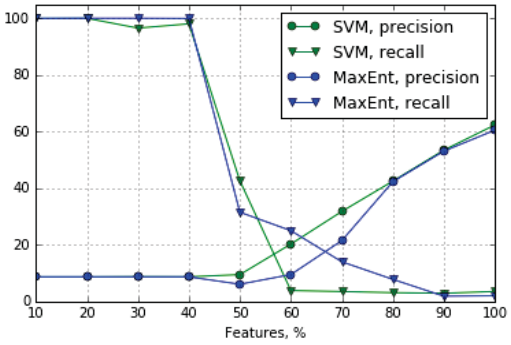

Figure 3. The effect of feature selection based on CHI on precision and recall of SVM and MaxEnt.

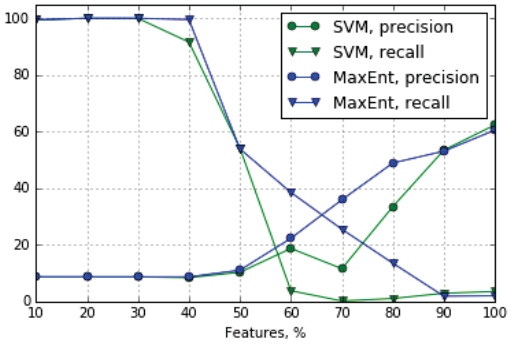

Figure 4. The effect of feature selection based on IG on precision and recall of SVM and MaxEnt.

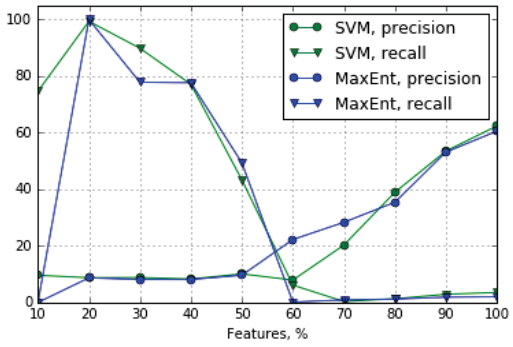

Figure 5. The effect of feature selection based on IGR on precision and recall of SVM and MaxEnt.

6.3 Feature Selection

In the next experiment, we examined the ability of the Chi-Square, Information Gain and Information Gain Ratio to select the most useful classification features. Computing scores for all the features, we experimented with keeping the top 10%, 20%, …, 90% of the most informative features. These results are shown in Figures 3, 4, and 5.

We see that the effect of feature selection is largely similar for CHI, IG and IGR. When features are removed drastically (keeping 40% of features or less), the recall improves all the way to 100%, while precision drops to about 10%. The fewer features are removed, the greater precision and the lower recall, with the best precision achieved when keeping 100% of features, both classifiers and all three feature ranking functions.

7 Conclusion

This is the first work to conduct a large-scale comparative evaluation of classification features extracted from Twitter posts, using different learning algorithms and datasets representing a broad range of mass emergencies including both natural and technological disasters. Our key findings can be summarized as follows. We presented empirical results demonstrating that a machine learning classifier tested on data that represents mass emergency events that were unseen at the training stage suffers a significant performance drop, especially in terms of recall, in comparison to testing on data that represents the same types of emergency events as the train data. We furthermore find that when testing the classifier on unseen event types, lexical features help to achieve better precision, while semantic, stylistic, and features derived from message metadata help improve recall. Finally, we examined several well-known feature selection methods, finding that they all produce a similar effect on the classifier: at aggressive levels of feature selection, they lead to better recall; however, they do not help much with precision.

This work has thus produced results that can inform development of applications for automatic detection of social media posts relating to mass emergencies, with regards to the

choice of features to be used under different use cases. Future research will focus on ways to exploit the described properties of the features: for example, create different feature subsets constituting different "views" on the data within a semi-supervised learning method.

8 References

[1] Zahra Ashktorab, Christopher Brown, Manojit Nandi, and Aron Culotta. 2014. Tweedr: Mining Twitter to inform disaster response. In Proc. of ISCRAM.

[2] Hila Becker, Mor Naaman, and Luis Gravano. 2011. Beyond trending topics: Real-world event identification on Twitter. Proc. of ICWSM. 438–441.

[3] Cornelia Caragea, Nathan McNeese, Anuj Jaiswal, Greg Traylor, H. Kim, Prasenjit Mitra, Dinghao Wu, A. Tapia, Lee Giles, Bernard J. Jansen, and others. 2011. Classifying text messages for the Haiti earthquake. In Proc. of ISCRAM.

[4] Muhammad Imran, Shady Elbassuoni, Carlos Castillo, Fernando Diaz, Patrick Meier. 2013. Extracting Information Nuggets from Disaster-Related Messages in Social Media. In Proc. of ISCRAM.

[5] Muhammad Imran, Carlos Castillo, Ji Lucas, Patrick Meier, and Sarah Vieweg. 2014. AIDR: Artificial intelligence for disaster response. In Proc. of WWW (Companion). IW3C2, 159–162.

[6] Karin Kipper, Anna Korhonen, Neville Ryant and Martha Palmer. Extending VerbNet with Novel Verb Classes. Proceedings of the Fifth International Conference on Language Resources and Evaluation -- LREC'06. May, 2006, Genoa, Italy: 2006.

[7] Rui Li, Kin Hou Lei, Ravi Khadiwala, and KC-C Chang. 2012. Tedas: A Twitter-based event detection and analysis system. In Proc. of ICDE. IEEE, 1273–1276.

[8] Benjamin Mandel, Aron Culotta, John Boulahanis, Danielle Stark, Bonnie Lewis, and Jeremy Rodrigue. 2012. A demographic analysis of online sentiment during hurricane Irene. In Proc. of the Second Workshop on Language in Social Media (LSM '12). Association for Computational Linguistics, Stroudsburg, PA, USA, 27-36.

[9] Chris Manning and Hinrich Schütze, Foundations of Statistical Natural Language Processing, MIT Press. Cambridge, MA: May 1999.

[10] Adam Marcus, Michael S. Bernstein, Osama Badar, David R. Karger, Samuel Madden, and Robert C. Miller. 2011. Twitinfo: Aggregating and visualizing microblogs for event exploration. In Proc. of CHI. 227–236.

[11] Aibek Musaev, De Wang, and Calton Pu. 2014. LITMUS: Landslide detection by integrating multiple sources. Proc. of ISCRAM.

[12] Alexandra Olteanu, Sarah Vieweg, Carlos Castillo. 2015. What to Expect When the Unexpected Happens: Social Media Communications Across Crises. In Proceedings of the ACM 2015 Conference on Computer Supported Cooperative Work and Social Computing (CSCW '15). ACM.

[13] Viktor Pekar, Naveed Afzal, and Bernd Bohnet. 2014. UBham: Lexical Resources and Dependency Parsing for Aspect-Based Sentiment Analysis. In Proc. of the Eighth International Workshop on Semantic Evaluation (SemEval 2014).

[14] Robert Power, Bella Robinson, John Colton, Mark Cameron. 2015. A Case Study for Monitoring Fires with Twitter. In Proceedings of the International Conference on Information Systems for Crisis Response and Management (ISCRAM'15).

[15] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes Twitter users: Real-time event detection by social sensors. In Proc. of WWW. ACM, 851–860.

[16] Vincent Schmidt and Jane Binner. 2011. A Semi-automated Display for Geotagged Text. Proceedings of the 2011 International Conference on Software Engineering Research and Practice.

[17] Fabrizio Sebastiani. Machine learning in automated text categorization. ACM Computing Surveys, 34(1):1-47, 2002.

[18] Tom De Smedt and Walter Daelemans. (2012). Pattern for Python. Journal of Machine Learning Research. 13, 2063-2067.

[19] Irina Temnikova, Carlos Castillo, and Sarah Vieweg. 2015. EMTerms 1.0: A Terminological Resource for Crisis Tweets. In Proceedings of the International Conference on Information Systems for Crisis Response and Management (ISCRAM'15).

[20] Sudha Verma, Sarah Vieweg, William J. Corvey, Leysia Palen, James H. Martin, Martha Palmer, Aaron Schram, and Kenneth Mark Anderson. 2011. Natural language processing to the rescue? Extracting "Situational Awareness" tweets during mass emergency. In Proc. of ICWSM.

[21] Hiroko Wilensky. 2014. Twitter as a Navigator for Stranded Commuters during the Great East Japan Earthquake, Proceedings of the 11th International ISCRAM Conference.

[22] Jie Yin, Andrew Lampert, Mark Cameron, Bella Robinson, and Robert Power. 2012. Using social media to enhance emergency situation awareness. IEEE Intelligent Systems 27, 6, 52–59.

[23] Andrea Zielinski and Ulrich Bügel. 2012. Multilingual Analysis of Twitter News in Support of Mass Emergency Events. In of the 9th International ISCRAM Conference – Vancouver, Canada, April 2012.