More Features Are Not Always Better: Evaluating Generalizing Models in Incident Type Classification of Tweets

Axel Schulz
Business Intelligence Marketing
DB Fernverkehr AG
Germany
schulz.axel@gmx.net

Christian Guckelsberger
Computational Creativity Group
Goldsmiths College, University of London
United Kingdom
c.guckelsberger@gold.ac.uk

Benedikt Schmidt
Telecooperation Lab, Technische Universität Darmstadt, Germany
benedikt.schmidt@tk.informatik.tu-darmstadt.de

Abstract

Social media represents a rich source of up-to-date information about events such as incidents. The sheer amount of available information makes machine learning approaches a necessity for further processing. This learning problem is often concerned with regionally restricted datasets such as data from only one city. Because social media data such as tweets varies considerably across different cities, the training of efficient models requires labeling data from each city of interest, which is costly and time consuming.

In this study, we investigate which features are most suitable for training generalizable models, i.e., models that show good performance across different datasets. We re-implemented the most popular features from the state of the art in addition to other novel approaches, and evaluated them on data from ten different cities. We show that many sophisticated features are not necessarily valuable for training a generalized model and are outperformed by classic features such as plain word-n-grams and character-n-grams.

1 Introduction

Incident information contained in social media has proven to frequently include information not captured by standard emergency channels (e.g. 911 calls, bystander reports). Therefore, stakeholders like emergency management and city administration can highly benefit from social media. Due to its unstructured and unfocused nature, automatic filtering of social media content is a necessity for further analysis. A standard approach for this filtering is automatic classification using a trained machine learning model (Agarwal et al., 2012; Schulz et al., 2013; Schulz et al., 2015b).

A problem for the classification approach is that language, style and named entities used in social media highly vary across different regions. Consider the following two tweets as examples: "RT: @People 0noe friday afternoon in heavy traffic, car crash on I-90, right lane closed" and "Road blocked due to traffic collision on I-495". Both tweets comprise entities that might refer to the same thing with different wording, either on a semantically low ("accident" and "car collision") or more abstract level ("I90" and "I-495"). With simple syntactical text similarity approaches using standard bag of words features, it is not easily possible to make use of this semantic similarity, even though it is highly valuable for classification.

These limitations impose constraints on the dataset, because tokens are likely to be related to the location where the text was created or contain location- or incident-sensitive topics. Models trained using spatially and temporally restricted data from one region are bound by the specific aspects of language and style expressed in the training data, thus, model reuse is not easily possible.

In this paper, we focus on the creation of generalized models. Such models avoid the use of features that — overfitting like — are only useful for a specific region. Generalized models are intended to work in different regions, even if training data originates only from one ore few regions. This can ensure high classification rates even in areas where only few training samples are available. Finally, in

421

times of increasing growth of cities and the merging with surrounding towns to large metropolitan areas, they allow to cope with the latent transitions in token use.

To create generalized models for incident type classification (and social media classification in general) the most important step is an appropriate feature generation. Therefore, in this paper we investigate the suitability of standard and novel features and different machine learning algorithms for the creation of generalized classification models for incident type classification. We conduct intensive feature engineering and evaluation. For this purpose, we have collected and labeled 10 datasets with high regional variation. To the best of our knowledge, this is the first investigation of the challenges of heterogeneous datasets in this domain, and of the suitability of state of the art classification and feature extraction techniques.

In summary, our contributions are: 1) Investigation of features and feature groups for generalized social media/incident type classification models. 2) Identification of the best feature combinations and classifiers for a generalized model. For an evaluation (qualitative and inferential statistics) of ten tweet datasets with high regional variation we get an overall F-measure of $> 83\%$. 3) The evaluation shows that features extending a plain n-gram-based approach are not necessarily valuable for training a generalized model as these provide little improvement.

Following this introduction, we give an overview of related work in Section 2. In Section 3, we provide a description of our datasets followed by a comprehensive evaluation in Section 4. We close with our conclusion and future work in Section 5.

2 Related Work

A review of existing work on the classification of social media content shows which features, feature groups and algorithms are generally used (see table 1). Furthermore, the number of classes and the dominating approaches unfold. We report the ratios of labeled tweets for the individual approaches; however, we omit performance measures as these are directly related to the respective datasets used for evaluation.

Classifiers based on Support Vector Machines (SVM) or Naive Bayes (NB) clearly dominate in terms of performance for incident type classi-

fication. (Sakaki and Okazaki, 2010; Carvalho et al., 2010; Agarwal et al., 2012; Robert Power, 2013; Schulz and Janssen, 2014) trained an SVM, whereas (Agarwal et al., 2012; Imran et al., 2013; Schulz and Janssen, 2014) also evaluated an NB classifier. In contrast to these works, (Wanichayapong et al., 2011) followed a dictionary-based approach using traffic-related keywords. (Li et al., 2012) do not provide any information about the classifier used.

Feature groups are mostly based on word-n-grams, such as unigrams (Carvalho et al., 2010), bigrams (Imran et al., 2013), or the combination of unigrams and bigrams (Robert Power, 2013; Karimi et al., 2013; Agarwal et al., 2012). (Schulz and Janssen, 2014) combined unigrams, bigrams, and trigrams. Also, based on the words present in the text named entities such as locations, organizations, or persons were used by (Agarwal et al., 2012; Li et al., 2012; Schulz and Janssen, 2014).

Twitter-specific features were also used, including the number of hashtags, @-mentions or web-text features such as the presence of numbers or URLs (Li et al., 2012; Agarwal et al., 2012; Robert Power, 2013; Karimi et al., 2013; Imran et al., 2013; Schulz and Janssen, 2014).

Keywords also play a crucial role in feature design. (Sakaki and Okazaki, 2010) used earthquake-specific keywords, statistical features (the number of words in a tweet and the position of keywords), and word context features (the words before and after the earthquake-related keyword). (Wanichayapong et al., 2011) used traffic-related keywords in combination with location-related keywords. Furthermore, Li et al. (2012) iteratively refined a keyword-based search for retrieving a higher number of incident-related tweets.

Two approaches rely on more specific feature groups. The approach of (Schulz and Janssen, 2014) is the only one that uses TF-IDF scores. (Imran et al., 2013) use Kipper et al.'s (2006) extension of the Verbnet ontology for verbs.

The related approaches mostly use word-n-grams and a variety of Twitter-specific features. Datasets are spatially and temporally restricted and limited to a small number, complicating generalizability.

3 Data Collection and Processing

We are interested in generalizable models for different regions, user-generated content has been

Table 1: Overview of related approaches for incident type classification. (NEs = Named Entities)

Approach	Classifier	#Classes	#Tweets	N-Grams	#NEs	#URLs	TF-IDF	Twitter	Other
(Sakaki and Okazaki, 2010)	SVM	2	597	x					Context
(Carvalho et al., 2010)	SVM	2	3,300	x					
(Wanichayapong et al., 2011)	Keyw.	2	1,249						
(Agarwal et al., 2012)	NB, SVM	2	1,400	x	x	x			
(Li et al., 2012)	Undefined	2	Undef.		x			x	
(Robert Power, 2013)	Keyw., SVM	2	794	x				x	
(Karimi et al., 2013)	SVM	6	5,747	x				x	
(Imran et al., 2013)	NB	3	1,233	x		x		x	Verbnet
(Schulz and Janssen, 2014)	SVM, NB	4	2,000	x	x	x	x	x	

created in. For this purpose, we created 10 datasets with more than 20k labeled tweets to train and test models with respect to their generalization. In the following, we describe how this data was collected, preprocessed, and which features were generated.

3.1 Data Collection

We focus on tweets as suitable example for unstructured textual information shared in social media. The classification of incident-related tweets represents a challenge that is relevant for many cities. We use a complex four-class classification problem, where new tweets can be assigned to the classes "crash", "fire", "shooting", and a neutral class "not incident related". This goes beyond related work with a focus on two-class classification. Our classes were identified as the most common incident types in Seattle using the Fire Calls data set (http://seattle.data.gov), an official incident information source.

As ground truth data, we collected several city-specific datasets using the Twitter Search API. These datasets were collected in a 15 km radius around the city centres of Boston (USA), Brisbane (AUS), Chicago (USA), Dublin (IRE), London (UK), Memphis (USA), New York City (USA), San Francisco (USA), Seattle (USA), Sydney (AUS).

We selected these cities because of their huge regional distance, which allows us to evaluate our approaches with respect not only to geographical, but also to cultural variations. Also, for all cities, sufficiently many English tweets can be retrieved. We chose 15 km as radius to collect a representative data sample even from cities with large metropolitan areas. Despite the limitations of the Twitter Search API with respect to the number of geotagged tweets, we assume that our sample is, although by definition incomplete, highly relevant to our experiments.

We collected all available Tweets during limited time periods, resulting in three initial sets of tweets: 7.5M tweets collected from November, 2012 to February, 2013 for Memphis and Seattle (**SET_CITY_1**); 2.5M tweets collected from January, 2014 to March, 2014 for New York City, Chicago, and San Francisco (**SET_CITY_2**); 5M tweets collected from July, 2014 to August, 2014 for Boston, Brisbane, Dublin, London, and Sydney (**SET_CITY_3**).

For the manual labeling process, we had to select a subset of our original tweet set which included our classes of interest for model training and testing. Generating subsets is required because manual labeling of social media data is very expensive, especially if multiple annotators are involved. To generate subsets we used the approach of (Schulz et al., 2013) of extracting microposts using incident-related keywords. As a result, more than 200 keywords were identified for each class. Based on these incident-related keywords, we were able to accurately and efficiently filter the datasets. After applying keyword-filtering, we randomly selected 5.000 microposts for each city. Though one might assume that this pre-filtering leads to a biased dataset, (Schulz and Janssen, 2014) showed that keyword sampling does not influence the classification process as the performance of a keyword-based classifier is notably worse compared to supervised classifiers.

In the next step, we removed all redundant tweets as well as those with no textual content from the resulting sets as a couple of tweets contain keywords that are part of hashtags or @-mentions, but have no useful textual content. The tweets were then labeled manually by five annotators using the CrowdFlower (http://www.crowdflower.com/) platform. We retrieved the manual labels and selected those for

Table 2: Class distributions for all datasets.

Dataset	Classes			
	Crash	Fire	Shooting	No
Boston	347	188	28	2257
Sydney	587	189	39	2208
Brisbane	497	164	12	1915
Chicago	129	81	4	1270
Dublin	131	33	21	2630
London	283	95	29	2475
Memphis	23	30	27	721
NYC	129	239	45	1446
SF	161	82	61	1176
Seattle	204	153	139	390

which all coders agreed to at least 75%. In the case of disagreement, the tweets were removed. This resulted in ten datasets with regional diversity to be used for evaluation.

Table 2 lists the class distributions for each dataset. The distributions vary considerably, allowing us to evaluate with typical city-specific samples. Also, the "crash" class seems to be the most prominent incident type, whereas "shootings" are less frequent. One reason for this is that "shootings" do not occur as frequent as other incidents, whereas another less obvious reason might be that people tend to report more about specific incident types and that there is not necessarily a correlation between the real-world incidents and those mentioned in tweets. Although the data sets have been filtered based on keywords, the "no incident" class remains the largest class.

One of the key questions that motivates our work is to which extent the used words vary in each dataset as an effect of the spatial and cultural context. We thus analysed how similar all datasets are by calculating the intersection of tokens. We found that after preprocessing, between 14% and 23% tokens are shared between the datasets. We do not assume that every unique token is a city-specific token, but the large number of tweets in our evaluations gives a first indication that there is a diversity in the samples that either requires the training of several individual- or one generalizing model which is the focus of this paper.

3.2 Preprocessing and Feature Generation

To use our datasets for feature generation, i.e., for deriving different *feature groups* that are used for training a classification model, we had to convert the texts into a structured representation by means of preprocessing. Following this, we extracted

several features for training classification models. To evaluate the best feature groups for incident type classification, we re-implemented commonly used feature extraction approaches from the state of the art. We further extended these feature groups by additional ones that seemed promising in this problem domain:

Preprocessing As a first step, the text was converted to Unicode to preserve non-Unicode characters. Specific URLs would not be useful for the classification process, therefore we replaced them with a common token "URL". We then removed stopwords and conducted tokenization. Every token was then analysed and non-alphanumeric characters were removed or replaced. Finally, we applied lemmatization to normalize all tokens. All preprocessing steps were performed by DKPro TC (de Castilho and Gurevych, 2014), a popular framework for text classification. After preprocessing, we generated several features (see Table 3). In the following, we give a description of the different feature groups.

Baseline Feature Sets As the most simple approach and as used in all related works, we represented tweets as a set of words and also as a set of characters with varying lengths. As features, we used a vector with the frequency of each n-gram. Most importantly, we evaluated the powerset of all different combinations of n-grams. For instance, if a length of $n = 2$ was chosen, we evaluated the three combinations $(n = 1), (n = 1, 2), (n = 2)$. Furthermore, as not all tokens are necessarily important for the classification process, we evaluated several top-k selection strategies, i.e., taking only the k most frequent n-grams into account. For this, we tested $k = 100, 1000, 5000$ as well as the approach using all n-grams. We treat these features as the baseline approach, and extend it by additional features, e.g. similarity, sentiment scores, Twitter-specific features.

Sentiment Features Emoticons are widely used to express emotions in textual content. Various text classification approaches make use of these, e.g. for sentiment analysis (Agarwal et al., 2011; Go et al., 2009). For incident type classification, they could also be useful as people link emotions with ongoing incidents, thus, we re-implemented three approaches for extracting sentiment features.

Table 3: Overview of all feature groups implemented for comparison

Feature Group	Description
Word-n-grams	Each tweet is represented as a powerset of word-n-grams of length $n = 1$ to $n = 3$.
Char-n-grams	Each tweet is represented as a powerset of char-n-grams of length $n = 1$ to $n = 5$.
POS_EMO	The Tweet NLP part-of-speech tagger (Owoputi et al., 2013) was used to identify emoticons. The ratio of emoticons to all tokens is calculated.
DICT_EMO	An emoticon library that is based on the suggestions from Agarwal et al. (Agarwal et al., 2011) was used comprising a set of 63 emoticons from Wikipedia. The number of positive and the number of negative emoticons in a tweet is calculated.
AGG_EMO	One single sentiment score based on the second approach by aggregating the number of positive and negative emoticons.
NER	We used the Stanford Named Entity Recognizer (Finkel et al., 2005) and applied the three class model to count the number of location, organization, and person mentions.
NR_CHAR	The number of characters in a tweet.
NR_SENT	The number of sentences in a tweet.
NR_TOKEN	The number of tokens in a tweet.
QUEST_RT	The proportion of question marks and sentences in a tweet.
EXCLA_RT	The proportion of exclamation marks and sentences in a tweet.
NR_AT_MN	The number of @-mentions in a tweet.
NR_HASHTAG	The number of hashtags in a tweet.
NR_URL	The number of URLs present in a tweet.
NR_SLANG	The number of colloquial words (i.e., *lol* or *ugh*). Feature extraction is based on the Tweet NLP POS-tags (Owoputi et al., 2013).
IS_RT	A boolean to indicate whether a tweet is a retweet.
NR_CARD	In conjunction with the named entities present in tweets, people tend to refer to street names (e.g.,*I-95*) or the number of injured people (e.g.,*2-people*). Thus, we create a feature for the number of cardinal numbers present in a tweet.
GREEDY_ST	Similarity scores following Greedy String Tiling (Wise, 1996) as a method to deal with shared sub-strings that do not appear in the same order.
LEVENST	The Levenshtein distance (Levenshtein, 1966) as an edit-distance metrics, i.e., the minimum number of edit operations that transform one tweet into another.
TF_IDF	As the baseline relies on plain frequency-based weighting, we calculate the traditional TF-IDF scores (Manning et al., 2009) for every tweet.

Named Entities: As shown in the state of the art, named entities, i.e. entities that have been assigned a name such as *Seattle*, are commonly used in tweets. Named entities might be valuable, as these are used frequently in incident-related tweets. Thus, we also incorporated Named Entity Recognition (NER) for feature extraction.

Stylistic Features: The style of a tweet could be an additional indicator for incident relatedness. For instance, a repetition of punctuations could point at a person that is expressing emotions resulting from an ongoing incident. Structured representation might indicate high quality.

Twitter-specific features As shown in related work, several Twitter-specific features seem to be valuable for incident type classification such as the number of @-mentions and hashtags.

Similarity Features The similarity of individual tweets might be helpful to identify common topics. We therefore implemented several similarity

scores[1]. The rationale behind this is to embrace additional features that do not only take the raw frequencies of words into account, but also which words appear in which document.

To sum up, we re-implemented two approaches that will serve as a baseline, and 18 additional feature groups to extend them. In the following section, we will evaluate the usefulness of these approaches for training a generalizing model.

4 Evaluation

The goal of our evaluation is to determine which features were most useful for creating a generalizing model. We first describe our method, including the feature sets, the classification algorithms used, and our sampling procedure. This is followed by a results section in which we report differences in performance by means of qualitative and inferential statistics.

4.1 Method

The indicators for well-performing features in related work allows us to perform a condensed evaluation, compared to similar studies such as (Hasanain et al., 2014).

Our approach comprises three steps: First, we evaluated the baseline approaches, i.e., word- and char-n-grams. Second, we combined each of the remaining features with the best performing baseline feature. Third, we again selected the best performing combinations and evaluated their power set. To evaluate the suitability of different features for training generalizing models, we picked one dataset from the 10 presented in Section 3.1 for training, and tested on the remaining 9 datasets. We did not evaluate different models on datasets from only one city, as we were interested in generalizing models.

Selecting each city as training set resulted in 90 performance samples per model. The models were formed by combining the feature sets described in the previous section 3.2 or respectively, their combinations, with an SVM and NB classifier. We decided for these classification algorithms since they were the most successful in related work. Another reason for the choice of NB is its good performance in text classification tasks, as demonstrated by Rennie et al. (2003). We relied on the LibLinear implementation of an SVM because it has been shown that for a large number of features and a small number of instances, a linear kernel is comparable to a non-linear one (Hsu et al., 2003). As for SVMs parameter tuning is inevitable, we evaluated the best settings for the slack variable c whenever an SVM was used. For training and testing, we used the reference implementations in WEKA (Hall et al., 2009).

We calculated the F1-Measure for assessing performance, because it is well established in text classification, cannot be manipulated by the classification threshold parameter and allows to measure

the overall performance of the approaches with an emphasis on the individual classes (Jakowicz and Shah, 2011). In Section 3.1, we demonstrated that the proportion of data representing individual classes varies strongly. We therefore weighted the F1-measure by this ratio and report the micro-averaged results over all datasets $\overline{F1}$. Given our focus on training a generalizable model, we deliberately did not focus on the performance variation in the individual datasets.

4.2 Results

In order to check whether our findings persist at least across the two learning algorithms, we did not aggregate the model performance samples but analyzed them for each algorithm separately. We therefore only have one independent variable, our feature groups, that affects the model performance. In order to keep p-value inflation low, we only compared the ten best performing models for each algorithm with respect to the F1-Measure. Note that even if the difference in performance between these models appears small, there are thus many worse models that were not explicitly listed.

Our samples generally do not fulfill the assumptions of normality and sphericity required by parametric tests for comparing more than two groups. Under the violation of these assumptions, non-parametric tests have more power and are less prone to outliers (Demsar, 2006). We therefore relied exclusively on the non-parametric tests suggested in literature: Friedman's test was used as non-parametric alternative to a repeated-measures one-way ANOVA, and Nemenyi's test[2] was used post-hoc as a replacement for Tukey's test.

In contrast to its parametric counterpart, Friedman's test is based on a ranking of the models induced by the performance measure, and therefore only relies implicitly on the latter. Each model is ranked from best to worst, with mean ranks being

---

[1]The respective similarity scores have been calculated on the whole document corpus after preprocessing.

[2]We chose Nemenyi's test because it is widely accepted in the machine learning community. A discussion of alternatives can be found in Herrera et al. (Herrera, 2008).

Feature Group	words(1000,1,2)	words(1000,1,3)	words(ALL,1,1)	words(5000,1,1)	words(100,1,1)	words(100,1,2)	words(100,1,3)	words(5000,1,3)	words(1000,1,1)	words(5000,1,1)
$\overline{F1}$	82.10	82.00	82.86	82.87	80.62	80.66	80.76	81.15	82.71	81.28

(a) LibLinear

Feature Group	words(1000,1,2)	words(1000,1,3)	words(1000,1,1)	words(5000,1,2)	chars(5000,2,3)	chars(5000,2,4)	chars(1000,2,4)	chars(1000,2,5)	chars(1000,2,3)	chars(5000,2,5)
$\overline{F1}$	80.10	79.56	80.10	78.09	78.01	80.27	80.22	79.73	79.86	80.48

(b) NaiveBayes

Table 4: Average F1-Measure $\overline{F1}$ for the ten best performing baseline feature groups

426

used in case of ties. The *Friedman statistic* is calculated by dividing the sum of squares of the mean ranks by the sum of squares error. For sufficiently many samples, the statistic follows a χ^2 distribution with $k-1$ degrees of freedom. The *q statistic* used in Nemenyi's test is similar to the one used by Tukey, but uses rank differences. It utilises the previous ranking from the Frieman test to calculate and relate the average ranks of two models, for each available pair. Two models are considered significantly different, if their difference in mean ranks exceeds a critical value, which varies for different significance levels. For a detailed description and examples of these tests, see (Jakowicz and Shah, 2011).

We illustrated the ranks and significant differences between the feature groups by means of the critical difference (CD) diagram. Introduced by Demsar (2006), this diagram lists the feature groups ordered by their rank, where lower rank numbers indicate higher performance. Feature groups are connected with bars if they are not significantly different, given $\alpha = 0.05$.

In the following, we will use shortcuts like *words(1000,1,2)* to denote the 1000 most frequent uni- and bigrams. The same applies for char-n-grams. Abbreviations can be found in Table 3.

4.2.1 Evaluation using LibLinear Classifier

We first evaluated which of our 20 baseline feature sets, as described in Section 3, lead to the best classification performance over different datasets. Notably, the ten best-performing approaches were all combinations of word-n-grams. Table 4 contains the average F-Measures for these approaches. The Friedman test indicated strong significant differences between the performances of these groups ($\chi_r^2(9) = 112.467, p < 0.001, \alpha = 0.01$). The subsequent Nemenyi test indicated strong significant pairwise differences between the performances of the models ($\alpha = 0.01$), with p-values listed in Table 2 in the supplementary.

Figure 1 illustrates the differences by means of a CD diagram: the approaches of using simple unigrams of the most frequent 5000 and all words provide the best results, i.e. they have the lowest rank. They are not significantly different from the 1000 most frequent word-uni and bigrams. Nevertheless, they are significantly better than all other baseline approaches.

This also applies to the char-n-gram approaches, that were not considered in this statistical comparison due to their inferior performance. It is important to note that the differences between the worst word-n-gram approaches and the best char-n-gram approaches could still be statistically non-significant.
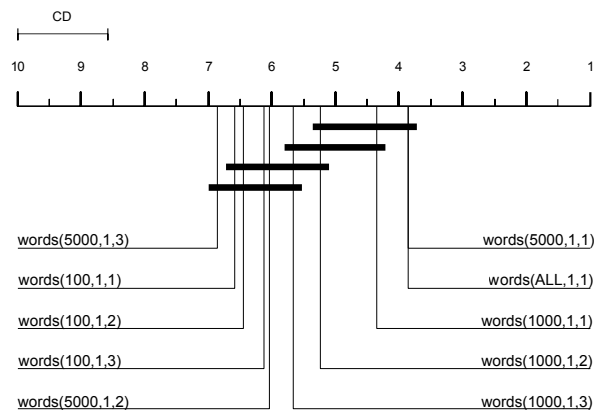

Figure 1: CD diagram with the ranks of the ten best performing baseline feature groups for LibLinear. Feature groups are connected if they are not significantly different ($\alpha = 0.05$).

The best performing baseline approach for LibLinear is using unigrams of the top 5000 words, i.e. *words(5000,1,1)*, with $\overline{F1} = 82.87$. We therefore picked this baseline feature group for the second part of our evaluation. We added each non-baseline feature individually to the selected baseline approach and compared the performances of these combinations and the non-extended baseline group. Table 6 lists the averaged F-Measure. When comparing the ten best-performing groups, the Friedman test showed strong significant differences between the performances of the models ($\chi_r^2(9) = 87.274, p < 0.001, \alpha = 0.01$). The Nemenyi test partly showed strong significant differences between the performances of the models (for the corresponding p-values see Table 3 in the supplementary). They are illustrated in the CD diagram in Figure 2. The tests indicate that adding *NER* and *NR_AT_MT* to the baseline approach provides the best performances with $\overline{F1} = 83.32$ and $\overline{F1} = 83.03$ respectively.

Finally, we evaluated the power set of these feature groups, i.e. we compared the individual groups and their combination. Table 5 contains the corresponding averaged F-Measures. For the resulting performance samples, the Friedman test showed strong significant differences between the models ($\chi_r^2(3) = 72.014, p < 0.001, \alpha = 0.01$). The Nemenyi test partly showed strong significant
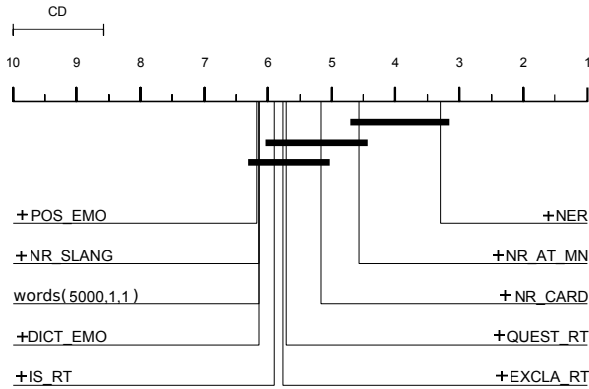
Figure 2: CD diagram with the ranks of the ten best performing feature groups for LibLinear, comprising the baseline and the baseline with an additional feature. Feature groups are connected if they are not significantly different ($\alpha = 0.05$).

Feature Group	words(5000,1,1)	+NER	+NER+NR_AT_MN	+NR_AT_MN
$\overline{F1}$	82.87	83.32	83.48	83.03

Table 5: Average F1-Measure $\overline{F1}$ for the power set of best performing feature groups and LibLinear.

differences ($\alpha = 0.001$), with p-values listed in Table 4 in the supplementary and illustrated in Figure 3. The diagram shows that the combination of *NER* and *NR_AT_MN* with the *words(5000,1,1)* baseline outperforms all other models with respect to F1 ($\overline{F1} = 83.48$), but does not differ significantly from the plain *NER* approach ($\overline{F1} = 83.32$). This combination gives us the final and best feature set for training a generalizing model over our datasets. As can be seen, the plain n-gram approach ($F1 = 82.87$) can be improved further by 0.5%.

4.2.2 Evaluation using Naive Bayes Classifier

In this section, we repeat the previous steps for the NB classifier. As baseline feature sets, we first evaluated the word-n-gram and char-n-gram approaches. The averaged F-Measures can be found in Table 4. The Friedman test showed strong significant differences between the performances of the models ($\chi_r^2(9) = 110.293, p < 0.001, \alpha = 0.01$). The Nemenyi test partly showed strong significant differences between the performances of the models (for the corresponding p-values see Table 1 in the supplementary). In contrast to the LibLinear classifier, using the 5000 most frequent combinations of two to five subsequent characters, i.e. *chars(5000,2,5)* provide the best F1 score ($\overline{F1} = 80.48$). Thus, char-n-grams outperform the
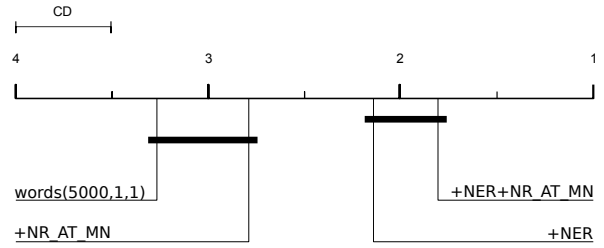

Figure 3: CD diagram with the ranks of the best baseline feature groups, complemented with a combination of the best performing feature sets, for LibLinear. Feature groups are connected if they are not significantly different ($\alpha = 0.05$).

word-n-gram approaches with respect to F1.

The CD diagram in Figure 5 shows that using either the 5000 most frequent char-n-grams with a length of two to five and two to four respectively, the 1000 most frequent word-n-grams with a length of one and one to two respectively, and the 1000 most frequent char-n-grams with a length of two to four do not differ significantly. However, using either the 5000 most frequent char-n-grams with a length of two to five and two to four respectively significantly outperform all other baseline approaches. As a subsequent step, we added each single feature to *chars(5000,2,5)* as the best baseline approach to find if these provide better performance for the NB classifier. Table 6 contains the corresponding averaged F-Measures. Though the Friedman test indicated strong significant differences between the performances of the models ($\chi_r^2(9) = 22.209, p = 0.008, \alpha = 0.01$), the subsequent Nemenyi test did not indicate significant pairwise differences. We can therefore not repeat the third step of our evaluation, but infer that for a NB classifier, the plain char-n-gram-based approach is sufficient for training a generalizing model for our dataset.

The results indicate that LibLinear provides a better avg. performance ($\overline{F1} = 83.32$) when training a generalizing model, compared to the NB classifier ($\overline{F1} = 80.48$).

5 Conclusion and Future Work

In this paper, we compared the performance of different popular feature groups and classification algorithms for the task of training a generalizing model for incident type classification. We carefully selected the most popular feature groups from related work, and separately evaluated them

428

Feature Group	words(5000,1,1)	+DICT_EMO	+NER	+NR_CARD	+NR_AT_MN	+POS_EMO	+NR_SLANG	+EXCLA_RT	+QUEST_RT	+IS_RT
$\overline{F1}$	82.87	82.87	83.32	83.06	83.03	82.87	82.87	82.88	82.88	82.88

(a) LibLinear

Feature Group	chars(5000,2,5)	+DICT_EMO	+QUEST_RT	+NER	+NR_AT_MN	+NR_HASHTAG	+POS_EMO	+NR_SLANG	+NR_SENT	+EXCLA_RT
$\overline{F1}$	80.48	80.48	80.49	80.55	80.51	80.48	80.48	80.48	80.48	80.50

(b) NaiveBayes

Table 6: Average F1-Measure $\overline{F1}$ for the ten best performing combinations of the best baseline and an additional feature
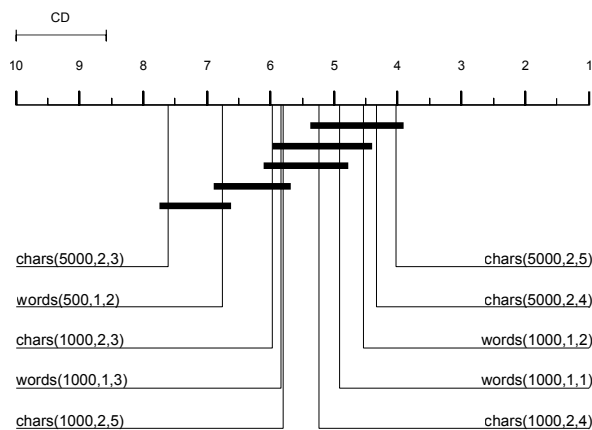

Figure 4: CD diagram with the ranks of the ten best performing baseline feature groups for Naive Bayes. Feature groups are connected if they are not significantly different ($\alpha = 0.05$).

Figure 5: Ranks of NB baseline feature groups.

for the LibLinear and NB classification algorithms on ten spatially and temporally diverse datasets. The resulting F1-measure samples indicate that training a generalizing model, i.e., a model that is applicable on previously unseen incident-related data, is still a challenging task. We found that LibLinear provides a better averaged performance compared to the NB classifier. More surprisingly, additional feature groups that are commonly used in related work do not necessarily outperform a plain n-gram-based approach. This highlights the need for other novel approaches for training generalizing classification models. Especially in the domain of incident detection and emergency management, our findings are important because less time consuming techniques showed nearly the same performance as sophisticated ones.

There are two main topics for our future work. First, we will investigate the performance of models generated with biased datasets on unfiltered datasets. This is relevant, if a technique like filtering is used to include more relevant class examples in a dataset than provided with an original sample – a necessary step to realize a labeled dataset for model learning of a rare-class task. Second, we will work on using novel features for the creation of generalized models. One example is the utilization of the Semantic Web to generate abstract features, utilizing a technique called Semantic Abstraction (Schulz et al., 2015a). Semantic Abstraction has shown to improve the generalization of tweet classification by deriving features from Linked Open Data and using location and temporal mentions.

References

Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. 2011. Sentiment analysis of twitter data. In *Proceedings of the Workshop on Languages in Social Media, LSM '11*, pages 30–38. ACL.

Puneet Agarwal, Rajgopal Vaithiyanathan, Saurabh Sharma, and Gautam Shroff. 2012. Catching the long-tail: Extracting local news events from twitter. In *Proceedings of the Sixth International Conference on Weblogs and Social Media, ICWSM 2012*. AAAI Press.

S. Carvalho, L. Sarmento, and R. J. F. Rossetti. 2010. Real-time sensing of traffic information in twitter messages. In *Proceedings of the 4th Workshop on Artificial Transportation Systems and Simulation ATSS, ITSC'10*, pages 19–22. IEEE Computer Society.

Richard Eckart de Castilho and Iryna Gurevych. 2014. A broad-coverage collection of portable nlp components for building shareable analysis pipelines. In *Proceedings OIAF4HLT at COLING 2014*, pages 1–11.

Janez Demsar. 2006. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, 7:1–30.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of ACL*, pages 363–370.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. In *Proceedings of the Workshop on Languages in Social Media, LSM '11*, pages 30–38. Association for Computational Linguistics.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18.

Maram Hasanain, Tamer Elsayed, and Walid Magdy. 2014. Identification of answer-seeking questions in arabic microblogs. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1839–1842. ACM.

Francisco Herrera. 2008. An Extension on Statistical Comparisons of Classifiers over Multiple Data Sets for all Pairwise Comparisons. *Journal of Machine Learning Research*, 9:2677–2694.

Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. 2003. A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University.

Muhammad Imran, Shady Elbassuoni, Carlos Castillo, Fernando Diaz, and Patrick Meier, 2013. *Extracting information nuggets from disaster- Related messages in social media*, pages 791–801. Karlsruher Institut fur Technologie (KIT).

Nathalie Jakowicz and Mohak Shah. 2011. *Evaluating Learning Algorithms. A Classification Perspective*. Cambridge University Press, Cambridge.

Sarvnaz Karimi, Jie Yin, and Cecile Paris. 2013. Classifying microblogs for disasters. In *Proceedings of the 18th Australasian Document Computing Symposium, ADCS '13*, pages 26–33. ACM.

Karen Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. 2006. Extending verbnet with novel verb classes. In *Proceedings LREC'06*.

VI Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707.

Rui Li, Kin Hou Lei, Ravi Khadiwala, and Kevin Chen-Chuan Chang. 2012. Tedas: A twitter-based event detection and analysis system. In *Proceedings of the 28th International Conference on Data Engineering, ICDE'12*, pages 1273–1276. IEEE Computer Society.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze., 2009. *An Introduction to Information Retrieval*, pages 117–120. Cambridge University Press.

Olutobi Owoputi, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A. Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *In Proceedings of NAACL*.

Jason D. Rennie, Lawrence Shih, Jaime Teevan, and David R. Karger. 2003. Tackling the poor assumptions of naive bayes text classifiers. In Tom Fawcett and Nina Mishra, editors, *International Conference on Machine Learning (ICML-03)*, pages 616–623. AAAI Press.

David Ratcliffe Robert Power, Bella Robinson. 2013. Finding fires with twitter. In *Australasian Language Technology Association Workshop*, pages 80–89. Association for Computational Linguistics.

Takeshi Sakaki and M Okazaki. 2010. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World Wide Web, WWW '10*, pages 851–860. ACM.

Axel Schulz and Frederik Janssen. 2014. What is good for one city may not be good for another one: Evaluating generalization for tweet classification based on semantic abstraction. In CEUR, editor, *Proceedings of the Fifth Workshop on Semantics for Smarter Cities a Workshop at the 13th International Semantic Web Conference*, volume 1280, pages 53–67.

Axel Schulz, Petar Ristoski, and Heiko Paulheim. 2013. I see a car crash: Real-time detection of small scale incidents in microblogs. In *ESWC'13*, pages 22–33.

Axel Schulz, Christian Guckelsberger, and Frederik Janssen. 2015a. Semantic abstraction for generalization of tweet classification: An evaluation on incident-related tweets. In *Semantic Web Journal: Special Issue on Smart Cities*.

Axel Schulz, Benedikt Schmidt, and Thorsten Strufe. 2015b. Small-scale incident detection based on microposts. In *Proceedings of the 26th ACM Conference on Hypertext and Social Media*. ACM.

N. Wanichayapong, W. Pruthipunyaskul, W. Pattara-Atikom, and P. Chaovalit. 2011. Social-based traffic information extraction and classification. In *Proceedings of the 11th International Conference on ITS Telecommunications, ITST'11*, pages 107–112. IEEE Computer Society.

Michael J. Wise. 1996. Yap3: Improved detection of similarities in computer program and other texts. In *SIGCSEB: SIGCSE Bulletin (ACM Special Interest Group on Computer Science Education*, pages 130–134. ACM Press.