Evaluating Multi-label Classification of Incident-related Tweets

Axel Schulz*+ Eneldo Loza Mencía+ Thanh Tung Dang† Benedikt Schmidt*

*Telecooperation Lab
Technische Universität Darmstadt
Germany

†Knowledge Engineering Group
Technische Universität Darmstadt
Germany

+HCI Research
SAP AG, Darmstadt
Germany

{aschulz,benedikt.schmidt}@tk.informatik.tu-darmstadt.de eneldo@ke.tu-darmstadt.de thanh.tung.dang@sap.com

ABSTRACT

Microblogs are an important source of information in emergency management as lots of situational information is shared, both by citizens and official sources. It has been shown that incident-related information can be identified in the huge amount of available information using machine learning. Nevertheless, the currently used classification techniques only assign a single label to a micropost, resulting in a loss of important information that would be valuable for crisis management.

With this paper we contribute the first in-depth analysis of multi-label classification of incident-related tweets. We present an approach assigning multiple labels to these messages, providing additional information about the situation at-hand. An evaluation shows that multi-label classification is applicable for detecting multiple labels with an exact match of 84.35%. Thus, it is a valuable means for classifying incident-related tweets. Furthermore, we show that correlation between labels can be taken into account for these kinds of classification tasks.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information filtering*; I.2.6 [**Artificial Intelligence**]: Learning

General Terms

Algorithms, Experimentation

Keywords

Microblogs, Multi-label Learning, Social Media

1. INTRODUCTION

Social media platforms are widely used by citizens for sharing information covering personal opinions about various topics (e.g., politics) as well as information about events such as incidents. In the latter case, citizens act as observers and create valuable incident-related information. For instance, during incidents such as the Oklahoma grass fires and the Red River floods in April 2009 [29], or the terrorist attacks on Mumbai [4], useful situational information was shared on Twitter. Also, Ushahidi, a social platform used for crowd-based filtering of information [15], was heavily used during the Haitian earthquake for labeling crisis-related information.

However, the discovery of incident-related information is a complex task, requiring the separation of valuable information from daily chatter in the vast amount of information created on social platforms. This can be realized based on techniques from data mining and machine learning. Classification is one method which can be utilized to extract relevant information from social networks (for tweets, see [23]). In a classification task, a system learns to label messages with exactly one label out of a predefined label set (e.g., "fire" or "crash"). This task is known as multi-class classification and widely used for text classification. However, during our research we found that assigning only one label would result in the loss of important situational information for decision making in crisis management. For instance, consider the following tweet:

> THIS CAR HIT THE FIRE HYDRANT AND CAUGHT FIRE....SOMEONE HOLIDAY ALTERED

A single label would necessarily lack relevant information. A better approach is the concurrent assignment of all three labels, which is known as multi-label learning. In the example, all labels ("fire", "crash", and "injuries") would be assigned concurrently using an appropriate learning algorithm. The example also shows that the assignment of multiple labels is not necessarily an independent process. Once the label for an incident type such as "crash" is assigned the probability of assigning the label "injuries" is changing. This dependency is known as label correlation and needs to be investigated in the context of multi-label learning.

With our analysis we want to investigate three important aspects of applying multi-label learning on incident-related tweets: (1) how to apply multi-label learners on tweets, (2) if the classification accuracy of multi-label classification approaches is comparable to the accuracy of multi-class classification approaches, and (3) if correlation between labels is a factor that needs to be taken into account for incident-related information. With this paper we contribute the first

Published as part of the #Microposts2014 Workshop proceedings, available online as CEUR Vol-1141 (http://ceur-ws.org/Vol-1141)

#Microposts2014, April 7th, 2014, Seoul, Korea.

in-depth analysis of multi-label classification of incident-related tweets. In summary, our contributions are twofold:

- We show that multi-label classification on incident-related tweets is applicable and able to detect the exact combinations of labels in 84.35% of the cases. Thus, we show that compared to common multi-class classification approaches, multi-label classification of incident-related tweets is a valuable means.
- We evaluate the influence of label correlation on the classification results of incident-related tweets. We show that for classification tasks label correlation needs to be taken into account.

The remainder of the paper is organized as follows. First, we describe and discuss related approaches. Second, the considered multi-label classification algorithms as well as the technical infrastructure (a machine learning pipeline) used for the analysis are presented. Next, we introduce our data collection setup and describe the evaluation of our approach. We close with a conclusion and future work.

2. RELATED WORK

Techniques of multi-label classification have been applied to domains such as text categorization [21, 13], music genre detection [20], or tag recommendation [7]. These application domains address long texts, images, or audio information. Text is probably one of the oldest domains in which the demand for categorization appeared, particularly multi-label categorization [25], with the first multilabel dataset (*Reuters-21578*) used in machine learning research being from the year 1987 [5, 8, 9]. Moreover, data is easily accessible and processable as well as vastly available. Hence, text classification was also one of the first research fields for multi-label classification and continues to be the most represented one among the commonly available benchmark datasets.[1]

A common application for texts is the classification of news articles [10, 18] for which the research focuses on scalability issues regarding the number of articles and especially the number of labels a text can be assigned to, which can sometimes go up to the thousands [11, 26]. News texts, as well as abstracts from scientific papers [14] or radiology reports [16] may sometimes be relatively short, but they are usually still structured and homogeneous. This kind of multi-label text classification problems were very well analyzed in the past and the used approaches showed to be effective (we refer the interested reader to the cited recent works).

In contrast, texts such as tweets are mostly unstructured and noisy, because of their limitations in size and the often used colloquial language. Related work on such short texts with a focus on solving multi-class problems exists, e.g., for sentiment analysis [24] or incident detection and classification [23]. In contrast to these approaches, this paper focuses on the use of multi-label classification for tweets.

Applying multi-label learning on very short texts is a topic of open research. Only two respective examples are known to the authors: Sajnani et al. [19] and Daxenberger et al. [1]. Sajnani et al. provided a preliminary analysis of multi-label classification of Wikipedia barnstar texts. Barnstars can be awarded by Wikipedia authors and contain a short textual explanation why they have been awarded. In this case, labels for seven work domains have to be differentiated. The authors show which features can be extracted from short texts for multi-label classification and evaluate several multi-label classification approaches. Daxenberger et al. categorize individual edits into non-exclusive classes like *vandalism*, *paraphrase*, etc.

Summarized, although many related approaches cope with multi-class classification of short texts such as microblogs, multi-label classification is an open research issue. Especially for the domain of crisis management, no prior research on this topic exists.

[1]Cf. `http://mulan.sourceforge.net/datasets.html` [28] and `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multilabel.html` repositories.

3. MULTI-LABEL CLASSIFICATION

In this section, we give an overview on multi-label classification. Multi-label classification refers to the task of learning a function that maps instances $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,a}) \in \mathcal{X} \subseteq \mathbb{R}^a$ to label subsets or label vectors $\mathbf{y}_i = (y_{i,1}, \ldots, y_{i,n}) \in \{0,1\}^n$, where $\mathcal{L} = \{\lambda_1, \ldots, \lambda_n\}$, $n = |\mathcal{L}|$ is a finite set of predefined labels and where each label attribute y_i corresponds to the absence (0) or presence (1) of label λ_i. Thus, in contrast to multi-class classification, alternatives are not assumed to be mutually exclusive, such that multiple labels may be associated with a single instance.

This makes multi-label data particularly interesting from the learning perspective, since, in contrast to binary or multi-class classification, there are label dependencies and interconnections in the data which can be detected and exploited in order to obtain additional useful information or just better classification performance. Some examples for our particular Twitter dataset were already shown up in the introduction. As we show, around 15% of our tweets could be assigned to more than one label, thus, we believe that it is not unusual to encounter tweets with several possible labels, so that in our opinion the view of microblogs as multi-labeled data seems more natural, more realistic, and more general. Nonetheless, previous work focuses on the multi-class labeling of tweets and this is the first work known to the authors which tries to exploit label dependencies on tweets.

In the following, we will describe commonly used approaches for multi-label classification: Binary Relevance (BR), Label Powerset (LP), and Classifier Chains (CC). All described techniques are based on the decomposition or transformation of the original multi-label problem into single-label binary problems, as most multi-label techniques do [27]. An illustration of these techniques is presented in Figure 1. This has the advantage that we can use state-of-the-art text classification algorithms for learning the binary problems such as support vector machines [25, 6]. We will also have a closer look at each classification approach with respect to taking dependencies between labels into account. Two of the used approaches are specifically tailored in order to cope with such dependencies.

3.1 Binary Relevance

The most common approach for multi-label classification is to use an ensemble of binary classifiers, where each classifier predicts if an instance belongs to one specific class or not. The union of all classes that were predicted is taken

$\mathbf{x}_i$	Labels $\in \{0,1\}^n$
$\mathbf{x}_1$	$(y_{1,1}, \ldots, y_{1,n})$
$\mathbf{x}_2$	$(y_{2,1}, \ldots, y_{2,n})$
$\vdots$	$\vdots \quad \ddots \quad \vdots$

(a) input training set

$\mathbf{x}_i$	Class $\in \{1, \ldots, 2^n\}$
$\mathbf{x}_1$	$\sigma(\mathbf{y}_1)$
$\mathbf{x}_2$	$\sigma(\mathbf{y}_2)$
$\vdots$	$\vdots$

(b) *label powerset* (LP) decomposition

$\mathbf{x}_i$	Class$_1 \in \{0,1\}$
$\mathbf{x}_1$	$y_{1,1}$
$\mathbf{x}_2$	$y_{2,1}$
$\vdots$	$\vdots$

$\cdots$

$\mathbf{x}_i$	Class$_n \in \{0,1\}$
$\mathbf{x}_1$	$y_{1,n}$
$\mathbf{x}_2$	$y_{2,n}$
$\vdots$	$\vdots$

(c) *binary relevance* (BR) decomposition

$\mathbf{x}'_i$	Class$_1 \in \{0,1\}$
$\mathbf{x}_1$	$y_{1,1}$
$\mathbf{x}_2$	$y_{2,1}$
$\vdots$	$\vdots$

$\cdots$

$\mathbf{x}'_i \in \mathbb{R}^a \times \{0,1\}^{n-1}$	Class$_n \in \{0,1\}$
$(\mathbf{x}_1, y_{1,1}, \ldots, y_{1,n-1})$	$y_{1,n}$
$(\mathbf{x}_2, y_{2,1}, \ldots, y_{2,n-1})$	$y_{2,n}$
$\vdots$	$\vdots$

(d) *classifier chains* (CC) decomposition

Figure 1: Decomposition of multi-label training sets into multiclass (LP) or binary (BR, CC) problems. $\mathbf{x}'_i$ denotes the augmented instance. During prediction, $y_{i,1}, y_{i,2}, \ldots$ in the extended input space is replaced by the predictions by $h_1^{CC}, h_2^{CC}, \ldots$ (see text).

as the multi-label output. This approach is comparable to classical one-against-all for a multi-class problem. Formally, we convert a training example pair $(\mathbf{x}_i, \mathbf{y}_i)$ into n separate pairs $(\mathbf{x}_i, y_{i,j})$, $j = 1 \ldots n$, one for each of the n base classifiers h_j. The predicted labels $\hat{y}_j$ for a test instance $\mathbf{x}$ are then the result of $h_j(\mathbf{x}) \in \{0,1\}$.

This method is fast and simple, however, it is not able to take label dependencies into account since each base classifier is trained independently from the other classifiers. As was recently stated by Dembczynski et. al [2], this is not necessarily a disadvantage if the objective is to obtain good label-wise predictions, such as measured by the Hamming loss (cf. Section 5). Therefore, BR serves as a fairly good performing baseline for our experiments.

3.2 Label Powerset

The basic idea of this algorithm is to transform multi-label problems into a multi-class classification problem by considering each member of the powerset of labels in the training set as a single class. Hence, each training example is converted into $(\mathbf{x}_i, \sigma(\mathbf{y}_i))$ with σ, σ^{-1} denoting a bijective function that maps between the label powerset of $\mathcal{L}$ and a set of 2^n meta-classes. The classifier h^{LP} is trained e.g. with one-against-all (like in our setting), and the prediction for $\mathbf{x}$ is obtained with $\sigma^{-1}(h^{LP}(\mathbf{x}))$.

LP takes label dependencies into account to some extent, as each distinct occurrence of a label pattern is treated as a new class. It is hence able to model the joint label distribution, but not explicitly and directly specific dependencies (correlations, implications, etc.) between labels. As a consequence, LP is tailored towards predicting exactly the correct label combination. As it is pointed out in [2] and contrary to what one may believe at first, this stays usually in contrast to predicting correctly each label individually (BR), i.e. we usually have a trade-off between both objectives.

In addition to the obvious computational costs problem due to the exponential grow of meta-labels, the sparsity of some label combinations, especially with an increasing number of labels, often causes that some classes contain only few examples. This effect can also be observed in our data, cf. Table 2.

3.3 Classifier Chains

As stated before in Section 1, it is very likely in our dataset that injured people are mentioned when also any incident type is mentioned (200 of 967 cases). On the other hand, it seems almost a matter of course that there was an incident if there is an injured person. Although this only happens in 200 out of 232 cases in our data we consider it relevant for larger data sets. The classifier chains approach (CC) of Read et al. [17] is able to directly capture such dependencies and has therefore become very popular recently.

The idea of this approach is to construct a chain of n binary classifiers h_j^{CC}, for which (in contrast to BR) each binary base classifier h_j^{CC} depends on the predictions of the previous classifiers $h_1^{CC} \ldots h_{j-1}^{CC}$. Particularly, we extent the feature space of the training instances for the base classifier h_j^{CC} to $((x_{i,1} \ldots x_{i,a}, y_{i,1} \ldots y_{i,j-1}), y_{i,j})$. Since the true labels y_i are not known during prediction, CC uses the predictions of the preceding base classifiers instead. Hence, the unknown y_j are replaced by the predictions $\hat{y}_j = h_j^{CC}(\mathbf{x}, \hat{y}_1 \ldots \hat{y}_{j-1})$.

This shows up one problematic aspect of this approach, namely the order of the classifiers in the chain. Depending on the ordering, CC can only capture one direction of dependency between two labels. More specifically, CC can only capture the dependencies of y_i on $y_1, \ldots, y_{i-1}$, but there is no possibility to consider dependencies of y_i on $y_{i+1}, \ldots, y_n$. Recovering our example from the beginning, we can either learn the dependency of the label *incident* given *injury* or the other way around, but not both. In addition, the effect of error propagation caused by the chaining structure may also depend on the label permutation. We will evaluate the effect of choosing different orderings for our particular dataset later on in Section 5.3.

Furthermore, CC has advantages compared to LP. CC is considered to predict the correct label-set, such as LP [2], but unlike LP, CC is able to predict label combinations which were not seen beforehand in the training data. In addition, the imbalance between positive and negative training examples is generally lower than for LP.

4. MULTI-LABEL CLASSIFICATION OF INCIDENT-RELATED TWEETS

In the following, the data used for multi-label classification of incident-related tweets is described in detail. The taken approach is composed of three steps. As a first step, unstructured text has to be converted into structured text. As a second step, the structured information needs to be transformed to features that can be used by a multi-label learner. Third, these features are used to train and evaluate a classifier.

4.1 Preprocessing of Unstructured Text

Our overall goal is to apply text mining on short docu-

ments that are present in social media, thus, they need to be represented by a set of features. As texts in social media are mostly unstructured, they first need to be converted into a representation which enables feature generation. Hence, as a first step, we apply Natural Language Processing. Firstly, we remove all re-tweets as these are just duplicates of other tweets and do not provide additional information. Secondly, @-mentions of Twitter users are removed from the tweet message as we want to prevent overfitting towards certain user tokens. Before further processing is applied, the text is converted to Unicode, as some tweets contain non-Unicode characters. Third, abbreviations are resolved using a dictionary of abbreviations based on the data provided by the Internet Slang Dictionary&Translator[2]. Then, we identify and replace URLs with a common token "URL". As a next step, stopwords are removed. This is important as very frequent words have limited influence when it comes to classifying tweets due to their relative frequency. Based on the resulting text, we conduct tokenization. Thus, the text is divided into discrete words (tokens) based on different delimiters such as white spaces. Every token is then analyzed and non-alphanumeric characters are removed or replaced. Also, lemmatization is applied to normalize all tokens. Additionally to the common NLP processing steps, we identify and replace location mentions such as "Seattle" with a common token to allow semantic abstraction. For this, we use the approach presented in [23] to detect named entities referring to locations (so-called location mentions) in tweets and to replace them with two tokens "LOC" and "PLACE".

4.2 Feature Generation

After finishing the initial preprocessing steps, we extracted several features from the tweets that are used for training a classifier. We conducted a comprehensive feature selection, analyzing the value of each feature for the overall classification performance. We compared word-n-grams, char-n-grams, TF-IDF [12] scores as well as syntactic features such as the number of explanation marks, question marks, and upper case characters. We found that the following features are the most beneficial for our classification problems:

- Word 3-gram extraction: We extract word three-grams from the tweet message. Each 3-gram is represented by two attributes. One attribute indicating the presence of the 3-gram and another attribute indicating the frequency of the 3-gram.
- Sum of TF-IDF scores: For every document we calculate the accumulated TF-IDF (term-frequency inverse-document-frequency) score [12] based on the single TF-IDF scores of each term in the document. The rational behind this is to create a similarity score which is not as strict as traditional TF-IDF scores, but allows forming of clusters of similar documents.
- Syntactic features: Along with the features directly extracted from a tweet, several syntactic features are expected to improve the performance of our approach. People might tend to use a lot of punctuations, such as explanation marks and question marks, or a lot of capitalized letters when they are reporting some incident. In this case, we extract the following features: the number of '!' and '?' in a tweet and the number of capitalized characters.
- Spatial features: As location mentions are replaced with a corresponding token, they appear as word unigrams in our model and can therefore be regarded as additional features.

4.3 Dataset

We focus on three different incident types throughout the paper in order to differentiate incident-related tweets. Three classes have been chosen, because we identified them as the most common incident types using the Seattle Real Time Fire Calls dataset[3], which is a frequently updated source for official incident information. We included also *injury* as an additional label. This results in four labels consisting of very common and distinct incident types and the injury label: Fire, Shooting, Crash, and Injury.

We collected public tweets in English language using the Twitter Search API, which provides geotagged tweets as well as tweets for which Twitter inferred a geolocation based on the user profile. For the collection, we used a 15km radius around the city centers of Seattle, WA and Memphis, TN. We focused on only two cities, as for our analyses we are interested in the stream of tweets for these cities and a specific time period instead of a scattered sample of the world, which could be retrieved using the Twitter Streaming API. This gave us a set of 7.5M tweets collected from 11/19/12 to 02/07/13. Though we know about the limitations of the Search API, we think that we collected a relevant sample for our experiments.

The dataset was further reduced to be usable for high quality labeling as well as the machine learning experiment. We first identified and extracted tweets mentioning incident-related keywords. Compared to other approaches that completely rely on filtering using hashtags, we take the whole message into account for identifying incident-related keywords. We retrieved a set of different incident types using the "Seattle Real Time Fire 911 Calls" dataset and defined one general keyword set with keywords that are used in all types of incidents like 'incident', 'injury', 'police', etc. For each incident type, we further identified specific keywords. For instance, for the incident type 'Motor Vehicle Accident Freeway' we use the keywords 'vehicle', 'accident', and 'road'. Based on these words, we use WordNet[4] to extend this set by adding the direct hyponyms. For instance, the keyword 'accident' was extended with 'collision', 'crash', 'wreck', 'injury', 'fatal accident', and 'casualty'. Based on these incident-related keywords, we filtered the datasets. Furthermore, we removed all re-tweets, as the originated tweets are also contained in our datasets and only these are needed for our experiments. Based on this filtered dataset, we randomly selected 20.000 tweets.

The selected tweets have been labeled manually by one researcher of our department. Out of these tweets, we randomly selected 2.000 tweets for further re-labeling for our multi-label classification problem. Those tweets were manually examined by five researchers using an online survey. To assign the final coding, we differentiated between two types of agreement:

[2] http://www.noslang.com

[3] http://data.seattle.gov

[4] http://wordnet.princeton.edu

Table 1: Overview of real-world incident types used for extraction of incident-related keywords as well as and the number of extracted keywords for keyword-based classification approach.

Class	**Fire**	**Shooting**	**Crash**	**Injury**
Real-World Incident Type	Fire In Building	Assault w/Weap	Motor Vehicle Accident	-
	Fire In Single Family Res	Assault w/Weapons Aid	Motor Vehicle Accident Freeway	
	Automatic Fire Alarm Resd		Medic Response Freeway	
	Auto Fire Alarm		Car Fire	
			Car Fire Freeway	
# of Keywords	148	36	73	23

Table 2: Distribution of the 10 label combinations occurring in the 2000 tweets of the dataset.

Label Combination	Number of Tweets
{}	971
{Fire}	313
{Shooting}	184
{Crash}	268
{Injury}	32
{Crash, Fire}	2
{Injury, Crash}	47
{Injury, Shooting}	149
{Injury, Fire}	33
{Injury, Fire, Crash}	1

- if four out of five coders agree on one label, only this label is assigned
- if less than four coders agree on one label, all labels which at least two coders assumed as correct are assigned as possible labels and further verified in a group discussion

The final labeled dataset consists of 10 different label combinations. The distribution for every combination is outlined in Table 2. The distribution indicates that around 15% (232) of all tweets in our dataset have been labeled with multiple labels. Another observation is that almost exactly 50% of the tweets do not have any label assigned, which is rather unusual compared to typically used and analyzed multi-label datasets[5]. In addition, the label cardinality, i.e., the average number of labels assigned to an instance, is around 0.59, whereas common datasets have at least more than 1 assigned. On the other hand, this is mainly due to the low number of total labels, since the label density (the average percentage of labels which are true) is 15%, which is a relatively high value. From a multi-label learning perspective, this is an interesting property of this dataset since it is not clear how commonly used techniques will behave under this circumstance. For example, many algorithms ignore instances without any label given.

[5]We refer to the repository at `http://mulan.sourceforge.net/datasets.html` for an overview of the statistics of the commonly used benchmark datasets in multi-label classification

5. EVALUATION

In the following section, we provide the evaluation results for the presented multi-label classification approaches on our dataset. We also present the result for a keyword-based approach as a simple way for conducting multi-label classification.

5.1 Evaluation Setup

We performed our experiments with Mulan, an open-source library for multi-label classification based on Weka [28]. We used two learners for our evaluation. First, we use the LibLinear implementation of support vector machines with linear kernel [3] as our base learner. We use the default settings, as we found that additional parameter optimization was not beneficial for improving the overall classification results. Second, we used the Weka implementation of Naive Bayes. The results were obtained using 10-fold cross validation.

The evaluation of multi-label problems requires different measures compared to those used for multi-class problems. In our paper, we use the following metrics:

Exact Match: Exact match is the percentage of the m test instances for which the labelsets were exactly correctly classified (with $[[z]]$ as indicator function returning 1 if z is true, otherwise 0)

$$ExactMatch(h) = \frac{1}{m}\sum_{i=1}^{m}[[y_i = h(\mathbf{x}_i)]] \quad (1)$$

Hamming Loss: The instance-wise Hamming loss [22] is defined as the percentage of wrong or missed labels compared to the total number of labels in the dataset. In this case, it is taken into account that an incorrect label is predicted and that a relevant label is not predicted. As this is a loss function, the optimal value is zero.

Recall, Precision and F1: We use micro-averaged precision and recall measures to evaluate our results, i.e., we compute a two-class confusion matrix for each label ($y_i = 1$ vs. $y_i = 0$) and eventually aggregate the results by (component-wise) summing up all n matrices into one global confusion matrix (cf. [27]). Recall and precision is computed based on this global matrix in the usual way, F1 denotes the unweighted harmonic mean between precision and recall. In Section 5, we also report recall, precision and F1 for each label using the label-wise confusion matrices.

5.2 Results for Keyword-Based Filtering

As mentioned before, we use a keyword-based pre-filtering for selecting an initial set of tweets that is suitable for la-

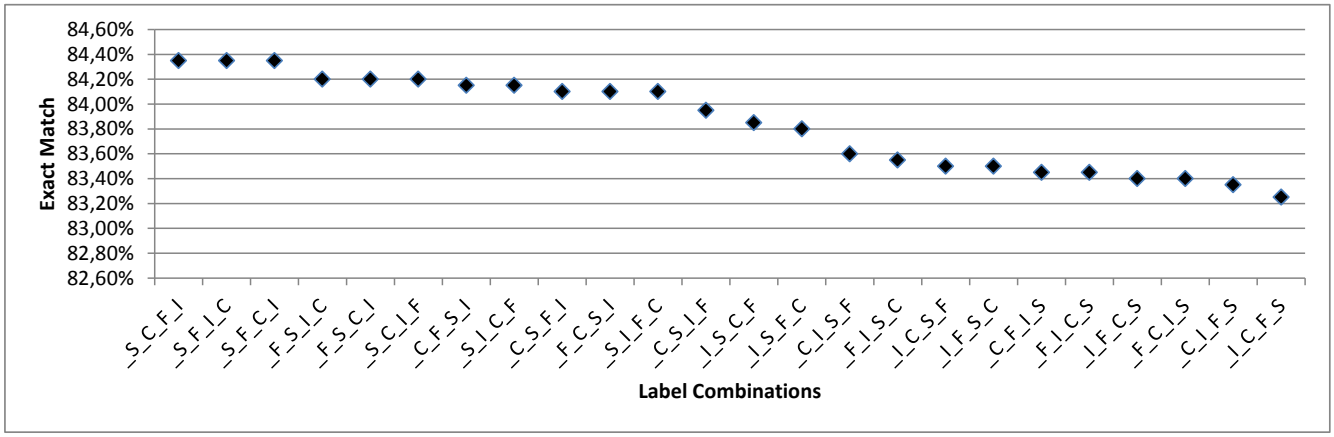

Figure 2: Percentages of exact matches for all label combinations.

beling. A first and simple approach for detecting incident related tweets is to use these keywords for classification.

In Table 1, the real-world incident types from the Seattle Real Time Fire Calls dataset and the corresponding number of extracted keywords is shown. For the injury class, no specific type in the Seattle dataset could be found, thus, we extended the set with a manually created list of keywords and their direct hyponyms.

The results for classifying each individual class are shown in Table 3. The results indicate that precision as well as recall are rather low. Only for the fire class a high recall could be achieved.

Table 3: Precision and recall for each individual label when applying keyword-based classification.

	Shooting	Fire	Crash	Injury
Precision	31.59%	54.12%	15.04%	63.64%
Recall	68.77%	95.99%	49.37%	37.40%

Furthermore, if the keywords would be used for applying multi-label classification, a precision of 32.22% and a recall of 64.90% is achieved, which is a rather bad result. Also exact match (28.45%) and h-loss (27.08%) are bad, thus, we conclude that with simple keyword-based filtering, multi-label classification cannot be done accurately.

5.3 Results for Multi-Label Classification

As a first step, we coped with the question if correlation between labels is taken into account and beneficial for the classification results. Thus, we evaluated all different label sequences using the classifier chains algorithm for our labels Fire (F), Shooting (S), Crash (C), and Injury (I). The values for exact match for each sequence are shown in Figure 2 (using SVM as our base learner).

The results indicate that the label sequence has indeed an influence on the classification performance. In our case, we get a difference of 1% between the best sequence Shooting, Crash, Fire, Injury and the worst Injury, Crash, Fire, Shooting. Also, we see that the Injury label is best used after incident labels have been classified - for the best cases even as one of the last labels in the sequence. It is also remarkable that classifying Shooting as first label followed up by either Crash or Fire is always a good option. This can be explained on the one hand by the generally good individual prediction performance for Shooting (cf., Table 5), hence leading to low error propagation, and on the other hand by the resulting label dependencies given the Shooting label is known: for instance, we can see from Table 2 that we can safely exclude Crash or Fire if there was a Shooting. This shows that our initial assumption that correlation between labels needs to be taken into account is true.

Based on the respective best (MAX) and the worst sequence (MIN), we compared CC to the multi-label approaches with the two different base learners. In Table 4 these evaluation results are shown. The first observation is that Naive Bayes is not adequate for classifying tweets, since though it achieves the best recall values using CC, this is in exchange of very low results on the remaining metrics and approaches. We will therefore focus on the results obtained by applying LibLinear as base learner. The results show that, if there is the opportunity of pre-optimizing the ordering of the labels, e.g., by performing a cross-validation on the training data, then classifier chains is able to slightly outperform the other approaches, which is most likely because the label correlation is valuable. This is also reflected in the good performance with respect to exact match, where the worst CC even outperforms LP, which is particularly tailored towards matching the exact label combination. Note also that LP is a common approach used for circumventing the need for a multi-label classification by creating meta-classes, as already mentioned in the introduction. However, this approach is always inferior to the compared approaches, which demonstrates the need for more advanced techniques in this particular use case.

We can also observe that improving the prediction of the exact label combinations may come at the expense of reducing the performance on label-wise measures, since the additional features used by CC generally lead to a higher potential deterioration (MIN) than potential improvement (MAX) for Hamming loss, recall, precision and F1, whereas for exact match this is not as clear.

As a last evaluation step, we evaluated the accuracy of each approach for every individual label. This is important as we want to understand how well a classifier performs for each label. The following Table 5 depicts the accuracy of individual labels using SVM with the best label order.

Table 4: Results for the different multi-label approaches and base learners obtained by cross-validation.

	Naive Bayes				SVM			
	BR	LP	CC - MIN	CC - MAX	BR	LP	CC - MIN	CC - MAX
Exact Match	59.60%	66.95%	71.15%	**72.45%**	83.85%	83.05%	83.25%	**84.35%**
H-Loss	15.02%	14.08%	9.400%	**9.175%**	4.688%	5.313%	4.900%	**4.588%**
F1	52.19%	55.37%	72.90%	**73.61%**	83.55%	81.53%	82.80%	**84.02%**
Precision	52.40%	55.34%	66.84%	**67.92%**	**93.61%**	90.28%	92.75%	93.46%
Recall	51.98%	55.39%	79.63%	**80.35%**	75.44%	74.35%	74.72%	**76.47%**

Table 5: Precision and recall for each individual label.

	BR (SVM)		LP (SVM)		CC (SVM)	
	Prec.	Recall	Prec.	Recall	Prec.	Recall
Shooting	95.7%	79.3%	92.0%	76.9%	95.7%	79.3%
Fire	94.7%	82.0%	90.3%	83.0%	93.3%	83.7%
Crash	90.8%	77.4%	88.0%	78.3%	90.9%	78.3%
Injury	92.9%	59.5%	91.1%	54.6%	93.0%	61.0%

The results show that the precision for individual labels is high with about 90% to 95% for each label, which is much better compared to the keyword-based classification. The differences between all approaches are nearly the same, thus, all approaches seem to be appropriate for classifying the individual labels. However, the recall drops significantly, depending on the label type. For instance, injuries often remain undetected. In this case, classifier chains show the best results for precision and recall. Note that the results for BR and CC on Shooting are the same, since the first classifier in the CC ordering is exactly trained like the corresponding BR classifier (cf. also Figure 1). This also shows that along the chain, CC slightly reduces the good precision of BR in exchange of improved recall.

5.4 Discussion

Though the results show the advantage of multi-label classification, we want to understand the limitations of our approach. Thus, we first created a confusion matrix for the classifier chains approach with the best label order. The matrix shows that most misclassifications occur due to an assignment of instances to the "no incident" label combination {}. The other wrong classifications are mostly a result of not detecting the injury label or of predicting it wrongly.

Table 6: Confusion matrix. The rows indicate the predicted/true label combinations and the columns the true/predicted ones.

	∅	F	C	F,C	I	F,I	C,I	F,C,I	S	F,S	I,S
∅	924	16	24	0	0	0	0	0	3	0	4
F	49	261	0	0	0	3	0	0	0	0	0
C	54	0	213	1	0	0	0	0	0	0	0
F,C	1	1	0	0	0	0	0	0	0	0	0
I	16	0	1	0	11	0	0	0	1	0	3
F,I	5	10	0	0	1	17	0	0	0	0	0
C,I	8	0	12	0	3	0	23	0	0	0	1
F,C,I	1	0	0	0	0	0	0	0	0	0	0
S	33	4	0	0	1	0	0	0	142	0	4
F,S	0	0	0	0	0	0	0	0	0	0	0
I,S	26	0	0	0	5	0	0	0	22	0	96

The following misclassified tweets show examples for such wrongly classified instances:

> "Tacoma Fire Department replaces 3 fire engines with pickup trucks: TACOMA Cutbacks within the Tacoma Fire... http://t.co/jPe2kuKG" ({} -> {F})

> "This girl is on fire. This girl is on fire. She's walking on fire. This girl is on fire - Alicia Keys #deep", ({} -> {S})

> "NeoMemphis News: Massive fire at factory in Ripley: Action News 5 is on the scene of a factory fire at ... http://t.co/brfnVbWp #memphis", ({F} -> {F,I})

The examples show that certain words such as "fire" or digits in the message might lead to wrong classifications. This could be avoided by adding additional features or with a larger training set.

In this section we have first shown that a simple keyword-based classification approach is not suitable for multi-label classification. Second, we presented results of state-of-the-art multi-label classification approaches and we showed that these perform quite well for classifying incident-related tweets. Compared to current approaches for the classification of microblogs, which rely on assigning only one label to an instance, the results show that it is possible to infer important situational information with only *one* classification step. The results also indicate that the label sequence has an influence on the classification performance, thus, this factor should be taken into account for following approaches.

6. CONCLUSION

In this paper we have shown how to apply multi-label learning on social media data for classification of incident-related tweets. Furthermore, we analyzed that we are able to identify multiple labels with an exact match of 84.35%. This is an important finding, as multiple labels assigned with one classification approach provide important information about the situation at-hand, which could not be easily derived from previously used multi-class classification approaches. Furthermore, we have shown that the natural relation of labels, which represents for instance the relation between incidents and injuries in the real-world, can be used and exploited by classification approaches in order to obtain better results.

For future work, we aim to add costs to our classifications. For instance, not detecting incident labels should be heavily punished compared to misclassifying the incident type. Furthermore, we aim to improve the overall performance of our approach by taking different features and a larger training set into account.

Acknowledgements

This work has been partly funded by the German Federal Ministry for Education and Research (BMBF, 01|S12054).

References

[1] J. Daxenberger and I. Gurevych. A corpus-based study of edit categories in featured and non-featured wikipedia articles. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, pages 711–726, Dec. 2012.

[2] K. Dembczynski, W. Waegeman, W. Cheng, and E. Hüllermeier. On label dependence and loss minimization in multi-label classification. *Machine Learning*, 88(1-2):5–45, 2012.

[3] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, aug 2008.

[4] R. Goolsby. Lifting Elephants: Twitter and Blogging in Global Perspective. In *Social Computing and Behavioral Modeling*. 2009.

[5] P. J. Hayes and S. P. Weinstein. CONSTRUE/TIS: A system for content-based indexing of a database of news stories. In A. T. Rappaport and R. G. Smith, editors, *Proceedings of the 2nd Conference on Innovative Applications of Artificial Intelligence (IAAI-90), May 1-3, 1990, Washington, DC, USA*, IAAI '90, pages 49–64. AAAI Press, Chicago, IL, USA, 1991.

[6] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In C. Nédellec and C. Rouveirol, editors, *Proceedings of 10th European Conference on Machine Learning (ECML-98)*, pages 137–142, Chemnitz, Germany, 1998. Springer-Verlag.

[7] I. Katakis, G. Tsoumakas, and I. P. Vlahavas. Multilabel text classification for automated tag suggestion. In *Proceedings of the ECML/PKDD-08 Workshop on Discovery Challenge*, Antwerp, Belgium, 2008.

[8] D. D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Devlopment in Information Retrieval*, pages 37–50, 1992.

[9] D. D. Lewis. Reuters-21578 text categorization test collection distribution 1.0. README file (V 1.3), May 2004.

[10] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. RCV1: A New Benchmark Collection for Text Categorization Research. *Journal of Machine Learning Research*, 5:361–397, 2004.

[11] E. Loza Mencía and J. Fürnkranz. Efficient pairwise multilabel classification for large-scale problems in the legal domain. In *Proc. ECML-PKDD-2008*, volume 5212 of *LNCS*, pages 50–65, Antwerp, Belgium, 2008. Springer.

[12] C. D. Manning, P. Raghavan, and H. Schütze. *An Introduction to Information Retrieval*, pages 117–120. Cambridge University Press, 2009.

[13] A. McCallum. Multi-label text classification with a mixture model trained by EM. In *AAAI'99 Workshop on Text Learning*, pages 1–7, 1999.

[14] A. Montejo Ráez, L. A. Ureña López, and R. Steinberger. Adaptive selection of base classifiers in one-against-all learning for large multi-labeled collections. In *Advances in Natural Language Processing, 4th International Conference (EsTAL 2004), Alicante, Spain, October 20-22, Proceedings*, volume 3230 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2004.

[15] O. Okolloh. Ushahidi, or 'testimony': Web 2.0 tools for crowdsourcing crisis information. *Participatory Learning and Action*, 59(January):65–70, 2008.

[16] J. P. Pestian, C. Brew, P. Matykiewicz, D. Hovermale, N. Johnson, K. B. Cohen, and W. Duch. A shared task involving multi-label classification of clinical free text. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, pages 97–104. Association of Computational Linguistics, June 2007.

[17] J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. *Machine Learning*, 85(3):333–359, June 2011.

[18] T. N. Rubin, A. Chambers, P. Smyth, and M. Steyvers. Statistical topic models for multi-label document classification. *Machine Learning*, 88(1-2):157–208, 2012.

[19] H. Sajnani, S. Javanmardi, D. W. McDonald, and C. V. Lopes. Multi-label classification of short text: A study on wikipedia barnstars. In *Analyzing Microtext*, 2011.

[20] C. Sanden and J. Z. Zhang. Enhancing multi-label music genre classification through ensemble techniques. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 705–714. ACM, 2011.

[21] R. E. Schapire and Y. Singer. Boostexter: A boosting-based system for text categorization. *Machine learning*, 39(2-3):135–168, 2000.

[22] R. E. Schapire and Y. Singer. BoosTexter: A Boosting-based System for Text Categorization. *Machine Learning*, 39(2/3):135–168, 2000.

[23] A. Schulz, P. Ristoski, and H. Paulheim. I see a car crash: Real-time detection of small scale incidents in microblogs. In P. Cimiano, M. Fernàndez, V. Lopez, S. Schlobach, and J. Völker, editors, *The Semantic Web: ESWC 2013 Satellite Events*, number 7955 in Lecture Notes in Computer Science, pages 22–33. Springer Berlin Heidelberg, 2013.

[24] A. Schulz, T. D. Thanh, H. Paulheim, and I. Schweizer. A fine-grained sentiment analysis approach for detecting crisis related microposts. In *Proceedings of the 10th International Conference on Information Systems for Crisis Response and Management (ISCRAM)*, pages 846 – 851, May 2013.

[25] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, Mar. 2002.

[26] G. Tsoumakas, I. Katakis, and I. P. Vlahavas. Effective and efficient multilabel classification in domains with large number of labels. In *Proceedings ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08)*, 2008.

[27] G. Tsoumakas, I. Katakis, and I. P. Vlahavas. Mining multi-label data. In O. Maimon and L. Rokach, editors, *Data Mining and Knowledge Discovery Handbook*, pages 667–685. Springer, 2010.

[28] G. Tsoumakas, E. Spyromitros Xioufis, J. Vilcek, and I. P. Vlahavas. Mulan: A java library for multi-label learning. *Journal of Machine Learning Research*, 12:2411–2414, 2011. Software available at http://mulan.sourceforge.net/.

[29] S. Vieweg, A. L. Hughes, K. Starbird, and L. Palen. Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems Pages (CHI'10)*, 2010.