Classification of Incident-related Tweets: Tackling Imbalanced Training Data using Hybrid CNNs and Translation-based Data Augmentation

Anna Kruspe*, Jens Kersten*, Matti Wiegmann*†, Benno Stein†, Friederike Klan*

*German Aerospace Center (DLR)
Institute of Data Science, Jena, Germany
<firstname>.<lastname>@dlr.de

†Bauhaus University
Weimar, Germany
<firstname>.<lastname>@uni-weimar.de

***Abstract*—In this paper, we present our four approaches submitted to the 2018 Text REtrieval Conference (TREC) Incident Streams (IS) track. One of the main challenges in this track is the lack of training data for certain classes defined in the ontology. We therefore take measures to expand the provided data; in a first step, additional tweets are manually selected from *CrisisLexT26* and *EMTerms* for all underrepresented classes, ensuring a minimum number of 50 tweets per class. Using this expanded data, we train four models. The first is a baseline model that uses a logistic regression classifier on word statistics. The second is a state-of-the-art CNN which considers different frame widths on pre-trained word embeddings. This model is then extended with two identical CNN branches trained on the *CrisisLexT26* and *CrisisNLP* data sets, and a posterior fusion network (third approach). Since all of these models still suffer from a lack of training data, more training examples are generated through a data augmentation technique using automatic round-trip translation. The fourth presented approach is identical to the third one, but is trained on this augmented data set.**

Finally, we describe our importance ranking procedure for tweets. Our method is implemented by weighting the average importance of the detected class and the tweet's relevance obtained with a classifier trained on the *CrisisLexT26* data set.

I. Introduction

The 2018 Text REtrieval Conference (TREC) Incident Streams (IS) track serves as an evaluation for the classification of tweets into incident-related classes. A class ontology, an annotated training data set, and a test data set without annotations were provided. The ontology comprises 25 classes describing a variety of topics during an incident, such as "Report-ServiceAvailable", "Other-Sentiment", "Request-SearchAndRescue", or "CallToAction-Donations". The training set contains around 1300 tweets related to 6 incidents, while the test data set is composed of around 22,000 tweets from 13 events. Submissions were expected to assign a class to each of these tweets as well as an importance score and a ranking within each event.

We focused on training fully automatic models in order to contribute to these tasks. This paper describes our four submissions to the challenge. We start by describing our data extension and augmentation procedures, then present our four classification approaches and a method for calculating the tweets' importance. Following this, we present an analysis of the results and finish with a small conclusion.

II. Data extension and augmentation

The main issue we came across while developing classification models was the selection of the training data. Both a class ontology and matching data were supplied for the challenge; however, some of the classes are underrepresented in the training data with just a handful of examples or, in extreme cases, none at all. Even the well-represented classes do not have the amount of training examples usually necessary for training classification models. For this reason, we first supplement the training data with manually selected tweets. These tweets are taken from the *CrisisLexT26* [5], [6] and *EMTerms* [7] data sets. We aimed at obtaining at least 50 examples per class. The ontology only provides rough descriptions of each class, and in many cases, there were not enough examples to obtain a clear idea what characteristics defined each class. In addition, many of the class definitions allow for overlapping annotations (e.g. a tweet could be both a "MultimediaShare" and a "FirstPartyObservation") or assume some sort of *a-priori* information (e.g. "KnownAlready") which leads to highly subjective judgement. These factors make the process of selecting additional training data challenging.

Manual selection is employed to obtain a base set of tweets for each class. Since this approach is tedious and costly, an automatic method was also developed for expanding the training data even further. This is done by running the existing tweets through an automatic translation engine to translate them into another language, then translating them back into English in the same way, introducing some lexical and semantic variety while keeping the meaning intact. This style of round-trip translation was first described by Lau et al. [1]. Ostyakov[1] then proposed employing it for the "Toxic Comments" Kaggle challenge[2], where Lee et al. won first prize with this approach[3]. Ostyakov implemented the translation

[1] https://github.com/PavelOstyakov/toxic

[2] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge

[3] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/discussion/52557

using the "TextBlob" Python library[4]. In contrast with this method, we do not perform the translation in a scripted manner; instead, we manually run chunks of the training data through Google Translate. This allows for a larger variety of translation languages, which are selected randomly from all available. In this way, we expand the amount of training examples per class to around 500.

III. Proposed models

In this section, we will describe the four models designed for classifying tweets into the classes provided in the ontology. The first of these models (*A*) is a baseline model primarily using logistic regression on frequency vectors; it is trained only on the original data. The other three (*B to D*) employ Convolutional Neural Networks (CNNs). The first two of these models are trained on the provided training data set plus the manually selected additional data. The last model (*D*) also uses the examples obtained with the augmentation procedure described above.

A. Logistic regression

In our first approach, several basic features are extracted from the tweets:

- The word 1-, 2-, and 3-gram frequencies, including stop words and only using terms that occur at least three times in total
- The character 1-, 2-, and 3-gram frequencies, including stop words and only using terms that occur at least three times in total
- The sentiment of the teaser message, as determined by Vader [8]
- The number of likes and the number of retweets
- Whether there is media attached and whether the user is verified

These features are fed into a logistic regression model, which is then trained on the 25 annotated classes. Only the original training data is used as some of the tweet metadata is not available for the additional data.

B. CNN

The first deep model that we tested is a Convolutional Neural Network (CNN) as proposed by Kim [3]. A visualization is provided in figure 1. This approach was specifically developed for classifying sentences into diverse categories, e.g. question types or sentiments. In contrast to the logistic regression model, the CNN only processes the tweet text data. At the input, the text data is transformed into an embedding using pre-trained weights. In the original model, this is done in two parallel channels: One with fixed embedding weights (static) and one which allows them to be adapted (non-static). Then, several convolutional layers with different kernel widths are applied in parallel. Global max-pooling is performed for each of these layers, and the results are concatenated. This new embedding is then fed into a fully connected layer with dropout to determine the final class. This type of model has successfully been used for crisis-related data [2].
Instead of generating an embedding from text data as in the original approach, we use a pre-trained embedding specific to crisis-related tweets [4] (only as a non-static channel). For the convolutional layers, kernel sizes of 3, 4, and 5 with 100 filters each are used, as suggested by Kim. Neither fixing the embedding nor adding filters or convolutional layers improve the results significantly.

C. Fusion CNN

According to preliminary manual inspection, the previously described CNN performs fairly well for the classification task, but suffers from the lack of training data for some classes, in addition to the unbalanced distribution between classes. This is expressed in a tendency to ignore, under-/overvalue, or overtrain on certain classes. Two approaches for overcoming this issue were implemented: A fusion CNN trained on the expanded data set, and a fusion CNN with data augmentation.
In a first idea, the model is supplemented with sub-models trained on the existing crisis-related tweet data sets *CrisisNLP* [4] and *CrisisLexT26*. These data sets both contain manual annotations. While the class ontologies used for annotating these data sets are not identical to that provided in the Incident Streams track, there is some overlap, as they refer to a similar problem statement. CNNs identical to the one described above (see section III-B) are trained for each of these data sets. Then, the outputs of these models are combined with that of the CNN trained directly on the TREC-IS data, and a fusion network is added to transform these results into a final class decision. For this step, not only the individual models' outputs (i.e. class probabilities) are taken into account, but also the outputs from the previous layer (i.e. a CNN embedding). The underlying idea here is that the two additional models will produce embeddings and intermediate classifications useful for solving the TREC-IS problem. Two versions of the CNN that is directly trained on the TREC-IS data are also integrated: One with the previously described crisis-specific work embeddings, and one with general-purpose GloVe embeddings[5]. The weights of the pre-trained networks are fixed, while the weights of the networks directly trained on TREC-IS and those of the fusion network are adapted during training. A schematic of this architecture is presented in figure 2.

D. Fusion CNN with augmented data

The second approach to overcome the lack of training data consists in the use of the augmented data described above (see section II). The described fusion network iss trained on a combination of the original training data, the manually selected additional tweets, and the examples automatically generated via round-trip translation.

[4] https://github.com/sloria/textblob

[5] https://nlp.stanford.edu/projects/glove/

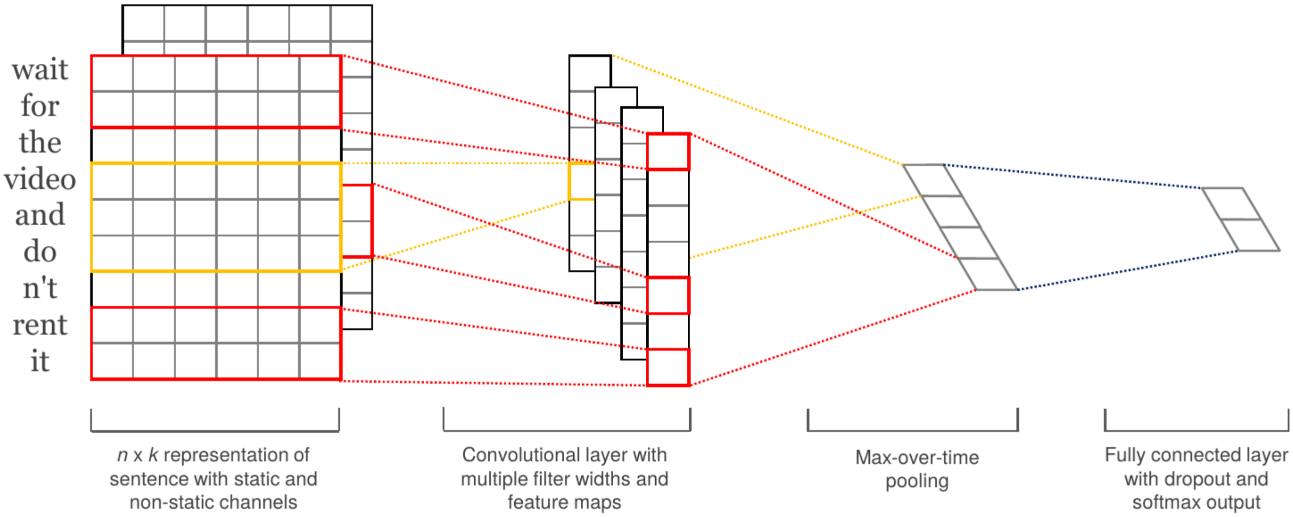

wait
for
the
video
and
do
n't
rent
it
n x k representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

Fig. 1: CNN for text classification as proposed by Kim [3].

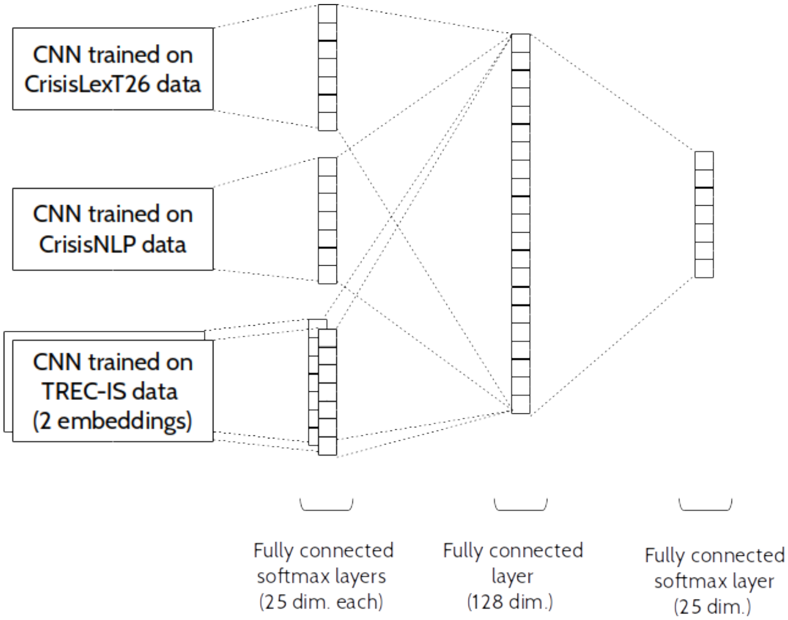

CNN trained on CrisisLexT26 data
CNN trained on CrisisNLP data
CNN trained on TREC-IS data (2 embeddings)
Fully connected softmax layers (25 dim. each)
Fully connected layer (128 dim.)
Fully connected softmax layer (25 dim.)

Fig. 2: A visualization of the fusion networks, consisting of sub-networks trained on *CrisisLexT26*, *CrisisNLP*, and TREC-IS data. The CNNs are identical to the one proposed in experiment *B* (also see figure 1).

IV. Importance scoring procedure

Submissions to the incident streams track also required an importance score for each test tweet, and a ranking according to these scores. No definition or criteria for such an importance scoring were provided. The training data annotations contain a "priority" field, although we are not sure if this corresponds to importance. We decided on a two-fold metric: One contributing factor is the *a-priori* importance v_c of each class, while the other is an invididual importance value v_i for each tweet. The individual importances v_i are obtained from another Yoon Kim CNN trained on the *CrisisLexT26* data set. In addition to class annotations, this data set also contains annotations regarding the individual "informativeness" of each tweet. This might be somewhat conceptually different from "importance", but we assume that there is a correlation. The weighted softmax likelihoods of the highest (inf_1) and second-highest (inf_2) importance class are taken into account. Their maximum is retained as an importance value specific to each individual tweet:

$$v_i = max(inf_1, 0.5inf_2) \tag{1}$$

On the other hand, class-wise importance v_c is calculated in two ways: First, by running the training data set through the described informativeness classifier and calculating the percentage of tweets per class that received the most "informative" label; second, by calculating the same percentage based on the "priority" information provided in the training data itself. Both results look relatively similar for most classes. Disparities occur for classes with a lack of training data; in these cases, values are chosen based on those of similar classes.

Finally, the harmonic mean of the importance attributed to the detected class and the informativeness obtained with the *CrisisLexT26* model is used as the final tweet score:

$$I = 2\frac{v_i * v_c}{v_i + v_c} \tag{2}$$

V. Results and Discussion

During evaluation, annotators were allowed to assign multiple classes to a tweet. For this reason, evaluation was performed in two modes: "Any-type", where a result was considered correct if the recognized class was part of the ground truth; and "Multi-type", where results were calculated on a 1-vs.-all basis. This means that for the second mode, systems only receive a score of 1/N for a correctly classified tweet where the annotator chose N classes, and can therefore not obtain perfect over-all scores.

The "any-type" and "multi-type" results are shown in figures 3a and 3b respectively. They allow for a number of interesting observations. Considering the "any-type" mode, the basic Yoon Kim CNN obtains the best recall and F1 values at 0.77 and 0.55, while the baseline logistic regression model has the best precision at 0.48. The more complex fusion CNN performs slightly worse, both when trained on the expanded and on the augmented data set. For the "multi-type" evaluation, the trend is different: The baseline model performs worst of all, while the three CNN models all have identical recall and F1 values. For precision, each addition to the CNN increases the result slightly. At first sight, this discrepancy is surprising. It becomes clearer, however, when considering that the "any-type" evaluation was essentially performed on a tweet-wise basis while the "multi-type" evaluation weights all classes equally. As described before, the classes are not equally distributed.

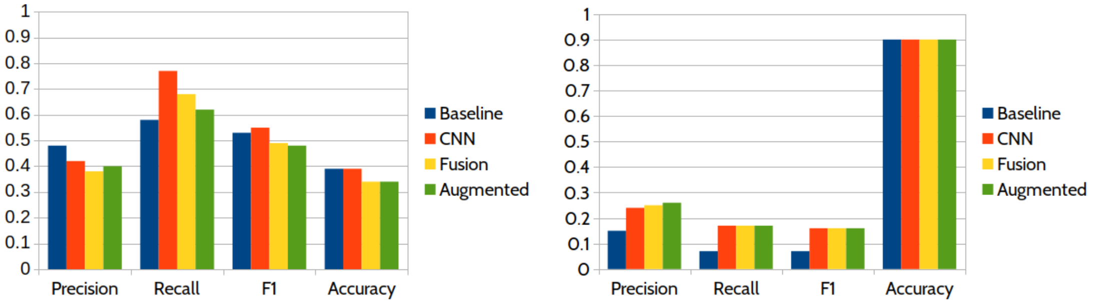
This effect can be further analyzed using the class-wise scores as shown in figure 4. In addition to the individual F1 values for each class, this plot also includes the number of original training examples available per class. The distribution of the evaluation examples is roughly similar. It becomes clear that the baseline model is strongly biased towards the frequent classes while ignoring the ones with few training examples. Since these classes also appear with high prevalence in the evaluation data, the model achieves high precision. A similar effect, although much weaker, can be observed for the basic CNN. In contrast, the fusion CNNs trained on both the expanded and the augmented training data perform somewhat worse for the overrepresented classes, leading to the lower results in the "any-type" evaluation. However, they often achieve better results on underrepresented classes, explaining the higher or equal "multi-type" results. Despite the over-all decrease, it is interesting to see that those models fulfil their purpose, making classes with little original training data more approachable. This could be an interesting direction for future research. The choice of model here is dependent on the goal of the final system - i.e., whether a high "any-type" F1 is desired or whether the model is expected to be able to detect underrepresented classes better.

Still, the over-all results leave a lot of room for improvement. Figure 4 also demonstrates that it is very hard to train the model for low-resource classes. In a production system, this problem would probably be solved by obtaining more training data, or possibly by accepting varying priors for the classes. As mentioned above, the ontology's chosen classes are not mutually exclusive, which is reflected in the evaluation strategy. A future model could be trained to perform multi-labeling as well.

Results for the importance scoring task are nearly identical for all four approaches at a mean squared error of 0.16. This is higher than the reported median error of all participants. Since our scoring procedures takes the recognized class into account, errors from this other task are propagated. In addition, ground-truth annotations were performed on a discrete scale instead of continuous values; in contrast, our scores represent a relative instead of an absolute measure of importance.

VI. Conclusion

For the 2018 TREC Incident Streams track, we submitted the results of four automatic text classification approaches. Our main contribution is the study of strategies to cope with the irregularities of the data: Some classes defined in the ontology have no or just very few corresponding examples in the



(a) Results for the "any-type" evaluation.

(b) Results for the "multi-type" evaluation.

Fig. 3: Over-all results for the four approaches.

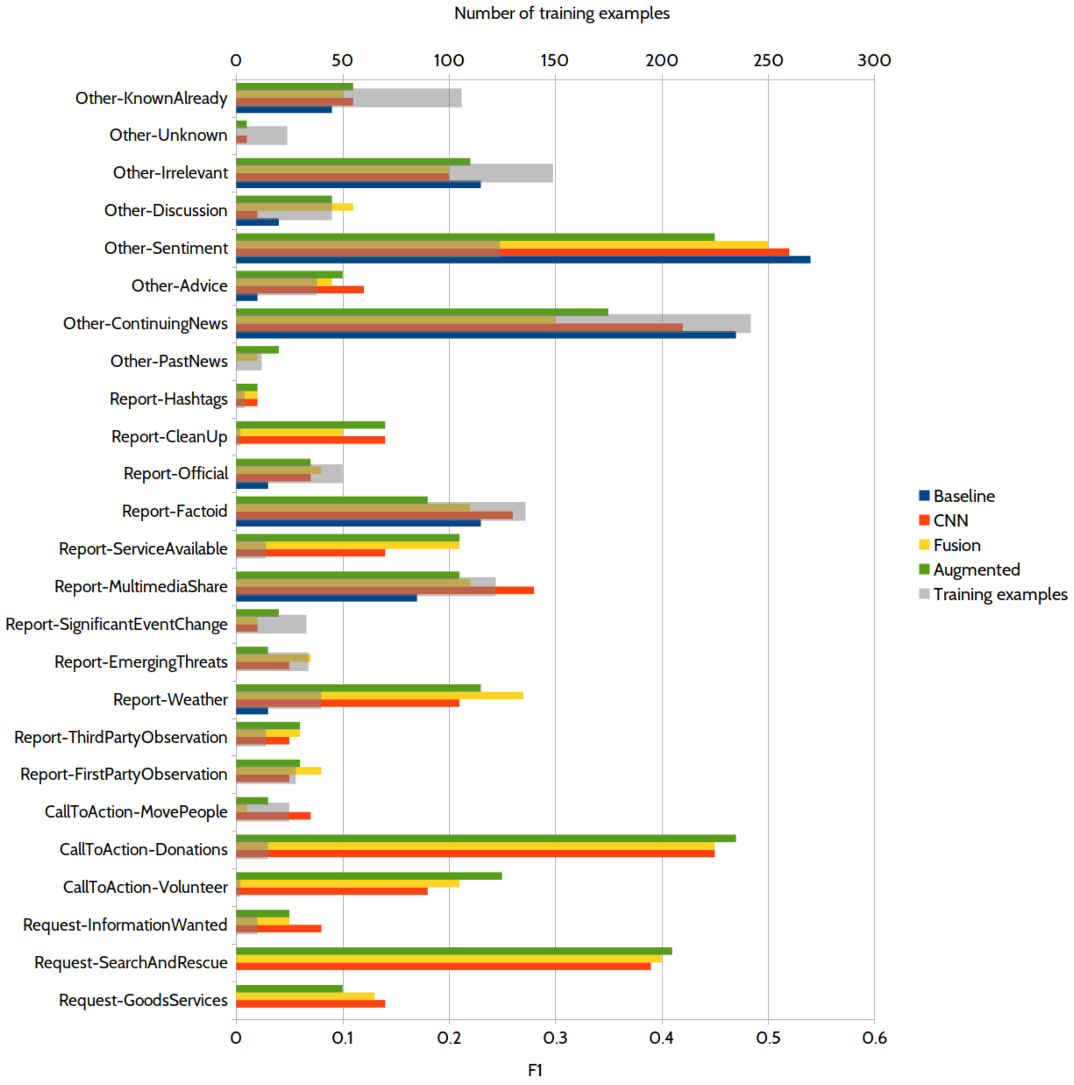


Fig. 4: Class-wise F1 measures for the four approaches (colored bars) and numbers of training examples per class (transparent gray bars).

training data. In addition, the class definitions are somewhat vague and overlapping. As a first remedy, we manually collect more training data from existing Twitter data sets.

We then institute a personal baseline for the Incident Streams task by training a logistic regression classifier on hand-crafted features based on word and character frequencies as well as tweet metadata, only using the original training data. Building on the state of the art in text classification, we also train a CNN as proposed by Kim on the expanded data. This CNN has been shown to perform well on other crisis-related data sets, but struggles here due to the limited training data. For this reason, we expand the architecture with two identical networks trained on existing crisis-related data sets. The three individual networks are consolidated with a subsequent fusion network. Finally, a data augmentation method employing round-trip translation is introduced. The same network as before is trained on this augmented data set.

We also propose a method for rating the importance of tweets with regard to incidents. This is done by taking into account both the *a-priori* importance of the detected semantic class and the informativeness of the individual tweet obtained with a model trained on a different data set.

Results show that the basic CNN performs best over-all at an F1 measure of 0.55. The fusion CNN approach, trained on the expanded and augmented data sets, demonstrates improvements for classes underrepresented in the original training set, which was the motivation for their development. Evaluation was performed in a different mode than training by allowing multiple annotations per tweet. A future system could, for example, improve on these results by also allowing this or by re-defining the classes, by using more training data, or by further developing these approaches.

References

[1] J. H. Lau, A. Clark, and S. Lappin, Unsupervised Prediction of Acceptability Judgements. In: 53rd Annual Conference of the Association of Computational Linguistics, July 2015, Beijing, China.

[2] G. Burel and H. Alani, Crisis Event Extraction Service (CREES) - Automatic Detection and Classification of Crisis-related Content on Social Media. In: 15th International Conference on Information Systems for Crisis Response and Management, May 2018, Rochester, NY, USA.

[3] Y. Kim, Convolutional neural networks for sentence classification. In: 2014 Conference on Empirical Methods in Natural Language Processing, October 2014, Doha, Qatar.

[4] M. Imran, P. Mitra, and C. Castillo, Twitter as a lifeline: Human-annotated twitter corpora for NLP of crisis-related messages. In: Tenth International Conference on Language Resources and Evaluation, May 2016, Portoroz, Slovenia.

[5] A. Olteanu, C. Castillo, F. Diaz, and S. Vieweg, CrisisLex: A Lexicon for Collecting and Filtering Microblogged Communications in Crises. In: AAAI Conference on Weblogs and Social Media, June 2014, Ann Arbor, MI, USA.

[6] A. Olteanu, S. Vieweg, and C. Castillo, What to Expect When the Unexpected Happens: Social Media Communications Across Crises. In: ACM 2015 Conference on Computer Supported Cooperative Work and Social Computing, March 2015, Vancouver, BC, Canada.

[7] I. Temnikova, C. Castillo, and S. Vieweg, EMTerms 1.0: A Terminological Resource for Crisis Tweets. In: International Conference on Information Systems for Crisis Response and Management, May 2015, Kristiansand, Norway.

[8] C. J. Hutto, E. Gilbert, VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. In: Proceedings of the Eighth International Conference on Weblogs and Social Media, June 2014, Ann Arbor, Michigan, USA.