CMU-Informedia at TREC 2019 Incident Streams Track

Junpei Zhou **Xinyu Wang**[*] **Po-yao Huang**[*] **Alexander Hauptmann**
Language Technologies Institute
Carnegie Mellon University
{junpeiz, xinyuw3, poyaoh, alex}@andrew.cmu.edu

Abstract

We describe CMU-Informedia's models for the TREC 2019 Incident Streams track. The goal of this track is classifying event/incident related tweets by High-level Information Types such as 'SearchAndRescue', 'InformationWanted' and so on. Each tweet should be assigned as many categories as are appropriate. What's more, this track requires predicting the Importance Scores, which is converted from the Importance Labels including 'Critical', 'High', 'Medium', 'Low' and 'Irrelevant'. For predicting the information types, we use feature extractors to extract features including meta-information, user entity, and textual embeddings, and then we build an information type predictor on those features. For predicting the importance scores, we build an importance score predictor which combines the scores derived from the predicted information types and the scores produced by a regression model. Evaluation results show that our models perform well on all metrics, and different models perform particularly well on different aspects.

1 Introduction

People often turn to social media when an emergency happens to find out relevant information. Twitter, as a platform for microblogging, can be especially useful for this purpose because it is designed for networking "what's happening in the world and what people are talking about right now." Twitter information has already been used for monitoring disasters and public safety events as in (Earle et al., 2012; Kumar et al., 2011; Poblete et al., 2018). Nevertheless, few studies have been done for filtering Twitter stream (Huang et al., 2018) down to actionable items, which can be particularly valuable to public safety personnel.

[*]equal contribution

To bring more efforts on categorizing information and aid requests made on social media, the TREC-IS task is introduced. The task (2019 edition) focuses on producing a series of curated multimodal feeds from tweets. Specifically, there are twenty-five high-level information types to be classified from tweets related to six incident types: bombing, earthquake, flood, typhoon/hurricane, wildfire and shooting. Importance scores reporting the priority for each tweet also need to be calculated.

Here we present the CMU-Informedia system for the TREC-IS task. The paper is organized as follows: in Section 2 we introduce relevant work to this task; in Section 3 we present the features used, the predictors for classifying information types and the predictors for computing importance scores; in Section 4 we describe the experiment setup and show experimental results; in Section 5 we draw the conclusions and indicate our future direction of work.

2 Related Work

The TREC-IS task was first introduced in 2018 (Mccreadie et al., 2019) to help emergency service operators monitor social media effectively. Due to the large volume and various modalities of information feed, how to categorize and verify them remains a challenging problem. The introduction of TREC-IS task helps bring more research efforts to addressing these practical problems.

In the 2018 edition, there are 1,335 tweets for training and nearly 20k tweets for testing. The BJUT (Lu et al., 2018) used an expansion module to augment the training set, as in the first edition there is not much training data. Then they trained a Support Vector Machine (SVM) with

TF-IDF word frequency features to classify the tweets. (Chy et al., 2018) built a rule-based classifier, an SVM classifier and a neural network classifier are ensembled for prediction. (Cumbreras et al., 2018) also used SVM and did topic expansion using either WordNet synonyms or word embeddings (Mikolov et al., 2013). Additional metadata is used to overcome the small training data in (Miyazaki et al., 2018). (Zahera et al., 2018) combined knowledge graph features and textual features and experimented with several traditional machine learning models. (Choi et al., 2018) represented the terms in tweets as conceptual entities such as event entities, category indicator entities, information type entities, URL entities, and user entities. Then they trained SVM models and deep learning models with class activation mapping for classification.

There are also other works using Twitter for disaster management. (Kumar et al., 2011) present a tool TweetTracker to monitoring and analyzing location and keyword specific tweets. (Ashktorab et al., 2014) build a Twitter-mining tool Tweedr to identify tweets reporting damage or casualties (Reitan et al., 2015) develop a method that can identify Japanese tweets refuting rumors which would hinder rescue activities during a natural disaster. (Huang et al., 2018) incorporate textual and visual information and learn visual-semantic embeddings (Huang et al., 2019) to monitor and filter out important tweets in public safety events. (Poblete et al., 2018) proposes an online method for detecting unusual bursts in discrete-time signals extracted from Twitter, which can be used for worldwide earthquake detection.

3 System

Our system has three parts: the feature extractor, the information types predictor, and the importance score predictor. The pipeline of our system is shown in Fig 1. In this system, the feature extractor extracts a lot of features and also pre-processes those features, then the information type is predicted by the information types predictor. What's more, the importance score is generated by the importance score predictor, which combines scores derived from the predicted information type and scores predicted by a regressor.

3.1 Feature Extractor

To get an informative representation for each tweet, we use several feature extractors. Some of them directly extract tweet-level features and some of them extract word-level features which are converted to tweet-level features later. There could be several choices in the implementation of each feature extractor. We will discuss the choices in this chapter and analyze our final choices with experimental results in Section 4.

Meta Information As we use the tweets extracted from the official Twitter API, the tweets we got contain a lot of meta-information, including some pre-processed entities and the count of some special actions such as 'retweet', 'favourite' and so on. Here are some features we used:

- The number of different entities (including 'hashtags', 'symbols', 'user mentions', 'urls' and 'media').
- The number of different actions (including 'retweet' and 'favorite').
- The polarity scores of sentiment analyzer from NLTK (Loper and Bird, 2002).
- The length of the tweet.
- Number of different characters.
- Number of words after tokenization.
- Number of words that have capitalized char, and the ratio of those words in this tweet.
- Some indicator features including if the tweet has coordinates, if the user is verified.
- Some user profiles including number of followers, friends, favourites, and statuses.

User Entity We have manually annotated about 200 users with mentions frequency greater than 10 in the training data. We followed (Choi et al., 2018) to define the six user entities: <UserNews>, <UserWeather>, <UserOrganization>, <UserDonation>, <UserDisasterInfo> and <UserMultimedia>. The user entities are annotated based on the user characterstics. For example, user @breakingstorm is annotated as <UserWeather> and user @NewEarthquake is annotated as <UserDisasterInfo>.

GloVe and FastText Apart from the meta information, we use the word embedding as a part of the feature for each word. There are two famous and popular word vector models: Global Vectors (GloVe) (Pennington et al., 2014) and FastText (Bojanowski et al., 2017; Joulin et al.,

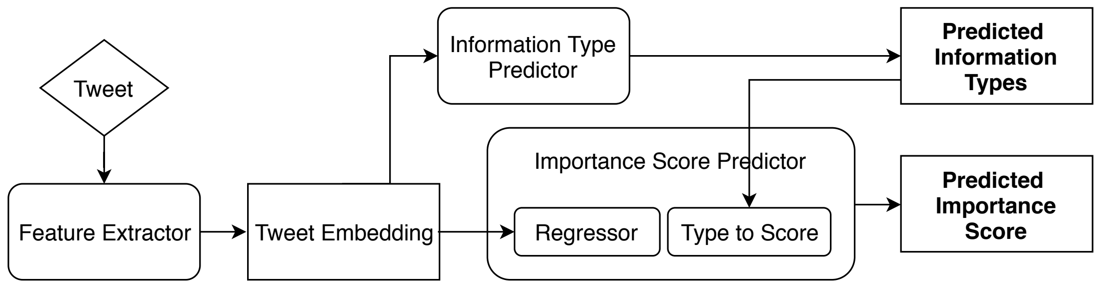
Tweet
Feature Extractor
Tweet Embedding
Information Type Predictor
Importance Score Predictor
Regressor
Type to Score
Predicted Information Types
Predicted Importance Score

Figure 1: The Pipeline of our system which finally output the information types and importance scores.

2016). GloVe uses a global co-occurrences matrix because the online scanning approach used by word2vec (Mikolov et al., 2013) does not fully exploit the global statistical information. FastText incorporates the subword information and it could be viewed as an extension of the word2vec model. We use the 200-dimension GloVe embedding pre-trained on Twitter 27 Billion corpus, and the 200-dimension pre-trained FastText model to extract the embedding for each word in the tweet. We also used a FastText model trained on emergency tweet corpus to overcome the out-of-domain issue. After extracting the features for each word, we tried two methods to derive the feature for the tweet, where the first is simple average, and the second is weighted average according to TF-IDF weights.

Skip-Thought The Skip-Thought (Kiros et al., 2015) model tries to extract a fixed-size embedding for a whole sentence. It similarly trains the model as word2vec, which means it uses the ordering of sentences in a natural language corpus as the training signal. We use the Skip-Thought model pre-trained on the BookCorpus (Zhu et al., 2015) dataset.

BERT In addition to training a context-free embedding as GloVe and word2vec, BERT (Devlin et al., 2018) trains a bi-directional language model as a feature extractor, which can distinguish the same word in different contexts. We use the pre-trained BERT model (Large, uncased) released by Google Research. For extracting features from BERT, there are a lot of choices, including using features from different layers, using features by taking the average or directly use the feature of the CLS symbol.

Preprocessing As the feature extracted by different models have different lengths and different scales, we need to merge them. The simplest method is to concatenate all features together, which is not the best choice because it ignores different scales and it leads to that the feature is too long to train in short time. So, we try to do some pre-processing for each feature, including the Principal Component Analysis (PCA) and normalization. Another choice is to do the late fusion, which means we don't concatenate those features in an early stage, but instead, we train models on each feature separately, and then merge the score predicted by different models.

3.2 Information Types Predictor

There are several changes in 2019 edition, so we made some adoptions accordingly in our model.

Multi-Type Categorization One of the most important differences compared with the setting in 2018 is that we need to do a Multi-Type Categorization (multi-label classification) for each tweet. It means we need to predict one or more Information Types for each tweet, and the ground truth also consists of several information types. Considering this setting, we use 'OneVsRestClassifier' wrapper in scikit-learn (Pedregosa et al., 2011) to train a classifier for each information type. In this task, we mainly focus on Naive Bayes, Linear SVM, Random Forest, and XGBoost (Chen and Guestrin, 2016). After getting the predicted score for each information type, we can use different strategies to predict the final information types. The first strategy is setting a threshold and then predict all labels whose scores are above that threshold. The second strategy is setting a k and choose the top-k scores and predictions.

Actionable Information Type Another change is that some information types are defined as 'actionable', such as 'GoodsServices', 'SearchAndRescue', 'MovePeople' and so on. The evaluation result distinguishes between performance on 'actionable' and 'non-actionable' information types. Our strategy is trying to give an additional weight w_a to those informative Information Types during training. The final weight w^i for each information type t_i is calculated by $w_a^i + w_b^i$ where w_b^i is the base weight for each information type, which is calculated according to the importance score associated with the t_i in the training set.

Event-wise Prediction We did some interesting explorations about using event-wise prediction. As all events fall into six incident types, there would be some traits for each event. For example, the tweets about wildfire should be different from the tweets about shooting. With this consideration, we train a model for each type of incident, and use the corresponding model during prediction.

Ensemble Another exploration we made is the ensemble, which combines the results predicted from different models. More specifically, after getting the prediction results from different models (such as Naive Bayes and Random Forest), we merge them to get the final result. However, the ensemble is not well-defined in the multi-label setting, so we try different methods for ensemble. The first method is voting, where we pick up the information types predicted by a majority of models. The second method is stacking, where we use the predicted score as new features, and train a new model on the development set.

3.3 Importance Score Predictor

The Importance Score is predicted by combining two methods. The first method is deriving the score from the information types we predicted, and the second method is trying to predict the score by a regression model directly.

Conversion from Information Type For each tweet, after getting the predicted information types from the Information Types Predictor, the score could be derived by averaging the score associated with each information type. The score for each information type is calculated from the training set (same as the base weight) as described in Section 3.2.

Regression The second method is using regression to predict the Importance Score. As the training data has a field of Importance Labels, it could be converted to the score according to the mapping as shown in Table 1. Then predicting the importance score could be easily modeled as a regression task, and we choose to use Ridge regression.

Label	Score	Label	Score
Irrelevant	0.0	Low	0.25
Medium	0.5	High	0.75
Critical	1.0		

Table 1: Mapping from Importance Label to Importance Score.

3.4 Four Models Submitted

As described in previous sections, there are a lot of choices in the implementation of each module. After exploring the performance by cross-validation, we finally choose four configurations to submit. There are a lot of trade-offs for choosing different strategies, for example, if we set a large additional weight for the actionable information type, the model could perform well on the actionable type, but its performance on other metrics might decrease. Those four models trained with different settings submitted by us focus on different aspects, and the detailed settings have been shown in Table 2. The 'Additional Weight' in the table means the additional weight we give to the actionable information types as described in Section 3.2. The performance of those four models on different metrics could be found in Section 4.3.

4 Experiments

For all tables in this section, 'HIAW' means High Importance Alert Worth, 'AAW' means Accumulated Alert Worth, 'HIITF1' means High Importance Information Type F1, 'ITF1' means Information Type F1, 'ITA' means Information Type Accuracy, 'PEEHI' means Priority Estimation Error High Importance, and 'PEE' means Priority Estimation Error.

4.1 Dataset

The size of each dataset is shown in Table 5, and during each submission, we use all available data, which including the training data of this task and the training along with testing data in previous editions.

4.2 Single-Label Experiments

When we participate in the 2019-A edition, we did experiments on 2018 labeled data for tuning hyperparameters. In the setting of 2018 edition, we only need to predict one label for each tweet, and it is much easier to implement those explorations

Submission Name	Model	Additional Weight	Importance Score
Informedia-rf1 (run1)	Random Forest	0.2	Only Regression
Informedia-rf2 (run2)	Random Forest	1.0	Half Prediction + Half Regression
Informedia-rf3 (run3)	Random Forest	5.0	Half Prediction + Half Regression
Informedia-nb (run4)	Naive Bayes	NA	Only Regression

Table 2: The configuration of different submissions.

Run id	HIAW	AAW	HIITF1	ITF1	ITA	PEEHI	PEE
run1	-0.9794	-0.4897	0.0300	0.1370	0.8638	0.1192	0.0665
run1-fix	-0.9197	-0.4609	0.0300	0.1390	0.8659	0.0815	0.0551
run2	-0.8323	-0.4224	0.0642	0.1283	0.8624	0.1025	0.0683
run3	-0.0898	-0.1837	0.0592	0.0568	0.8434	0.1660	0.2063
run4	-0.9197	-0.4609	0.1321	0.0995	0.8605	0.0788	0.0544
run4-fix	-0.9197	-0.4609	0.1559	0.1012	0.8601	0.0788	0.0544

Table 3: Official evaluation result for the four runs we submitted. The run id with '-fix' suffix is the result of the new hyper-parameter after making the metric in the cross-validation consistent with the official metrics.

Run id	HIAW	AAW	HIITF1	ITF1	ITA	PEEHI	PEE
run1	-0.9905	-0.4953	0.057	0.1309	0.8518	0.0855	0.0647
run2	-0.6196	-0.3338	0.0402	0.1136	0.8445	0.1307	0.0775
run3	0.2533	-0.2159	0.061	0.0731	0.8268	0.243	0.3798
run4	-0.9905	-0.4953	0.0372	0.1752	0.8449	0.0856	0.0648

Table 4: Evaluation result on 2019-A test data with settings of four runs we submitted.

Task Name	Training Size	Test Size
2018	1,335	22,216
2019-A	NA	6,134
2019-B	NA	13,916

Table 5: Number of tweets in each dataset after removing duplicates.

we mentioned in Section 3.2 with the scikit-learn framework. As the dataset is a little bit skewed, we use stratified K-fold cross-validation.

There are a lot of features that could be used as described in Section 3.1, so we did an ablation study (Table 6) to explore the influence of each feature by gradually adding features one by one. There are multiple candidates when it comes to choosing a model to do the classification, and we have explored the performance of them with different settings in PCA and event-wise prediction (Table 7). The evolving procedure of our model (Table 8) shows that tuning hyper-parameters and using event-wise prediction helps a lot, and the Random Forest is really powerful.

4.3 Multi-label Experiments

It is easy to implement stratified K-fold splitting in the one-label setting. However, it is not trivial to do so in the multi-label setting, and 'stratified-KFold' in scikit-learn doesn't support multi-label setting. To cope with it, we implement a stratified sampling algorithm for the multi-label data based on the work of Sechidis et al. (Sechidis et al., 2011), which can achieve nearly perfect ratio of each label, such as 4:1 for 5-fold sampling. To get a comparable result, we use the official V3 evaluation script released for TREC-IS 2019-A task.

For the event-wise prediction, it is interesting that this strategy brings improvements as shown in Table 7, but it hurts the performance significantly

Feature Used	Precision	Recall	F_1 score	Accuracy
MetaInfo	0.3875	0.6395	0.4825	0.3359
+ FastText	0.4057	0.6692	0.5051	0.3534
+ BERT	0.4144	0.6792	0.5147	0.3616
+ SkipThought	0.4257	0.6887	0.5261	0.3735

Table 6: Ablation Study for features on TREC-IS 2018 data for 'any valid type' and 'micro'.

System	Event-wise	PCA	F_1 score
NB	no	without	0.6768
NB	no	with	0.6936
NB	yes	without	0.7047
NB	yes	with	0.8763
SVM	no	with	0.6934
SVM	no	without	0.7496
RF	no	without	0.7775
RF	no	with	0.7845
XGBoost	no	without	0.8032
XGBoost	no	with	0.8062

Table 7: Explore the performance for different models with different settings. Here 'NB' means Naive Bayes, 'SVM' means Linear SVM, 'RF' means Random Forest.

System	F_1 score
NB with MetaInfo (Baseline)	0.4825
Use all features	0.5609
Use Event-wise	0.6146
Tune params of NB	0.6709
Use Event-wise for new params	0.7047
Use Random Forest	0.7783
Use Event-wise	**0.7961**

Table 8: The evolve history of our model guided by the performance on TREC-IS 2018 data ('NB' means 'Naive Bayes Model').

in the multi-label prediction (as shown in Table 9). It can be explained by the fact that for each type of event, the number of training data is much smaller, and some minor type of event even doesn't have all information types in its training set.

The official evaluation results on 2019-B test data for those four runs we submitted could be found in Table 3. The evaluation result on 2019-A test data for those four settings we submitted is shown in Table 4. We also explored the influence of the additional weights for actionable information types, and the experimental results on 2019-A test data could be found in Table 10, 11, and 12. 'Weight' means additional weight for the actionable information type during training.

5 Conclusion and Future Work

We have described CMU-Informedia's models for the TREC 2019 Incident Streams track. Our models achieve good performance for information type classification and demonstrate strong potential to identify important tweet feeds. We observe that the generalizability remains a challenging issue as the model-wise performance is inconsistent among tasks. We consider incorporating multimodal information and building a generalizable model as our future work.

Acknowledgement

The work reported here was supported in part through the financial assistance award 60NANB17D156 from U.S. Department of Commerce, National Institute of Standards and Technology and by DARPA grant FA8750-18-2-0018 funded under the AIDA program. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation herein. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA, DOI/IBC, or the U.S. Government.

References

Zahra Ashktorab, Christopher Brown, Manojit Nandi, and Aron Culotta. 2014. Tweedr: Mining twitter to inform disaster response. In *ISCRAM*.

Model	Event-wise	HIAW	AAW	HIITF1	ITF1	ITA	PEEHI	PEE
Naive Bayes	No	-0.9197	-0.4609	0.1321	0.0995	0.8605	0.1773	0.0758
Naive Bayes	Yes	-0.9774	-0.4887	0.0907	0.1228	0.8560	0.1939	0.1092
Random Forest	No	-0.9905	-0.4953	0.0570	0.1309	0.8518	0.1836	0.0913
Random Forest	Yes	-0.9674	-0.4857	0.0032	0.0696	0.8435	0.1831	0.1040

Table 9: The influence of event-wise prediction in multi-label experiments.

Weight	HIAW	AAW	HIITF1	ITF1	ITA	PEEHI	PEE
0.2	-0.9942	-0.4971	0.057	0.1309	0.8518	0.1836	0.0913
0.5	-0.7357	-0.4047	0.0404	0.1296	0.849	0.1556	0.0715
1.0	-0.466	-0.3201	0.0402	0.1136	0.8445	0.1393	0.0700
5.0	0.2533	-0.2197	0.0610	0.0731	0.8268	0.1565	0.1722
10.0	0.2795	-0.2412	0.0684	0.056	0.8213	0.1928	0.2500

Table 10: Evaluation results on 2019-A test data for Random Forest model with different additional weights using predicted information types to get importance scores.

Weight	HIAW	AAW	HIITF1	ITF1	ITA	PEEHI	PEE
0.2	-0.9942	-0.4971	0.057	0.1309	0.8518	0.1129	0.0615
0.5	-0.9942	-0.4972	0.0404	0.1296	0.849	0.1447	0.0642
1.0	-0.6196	-0.3338	0.0402	0.1136	0.8445	0.1307	0.0775
5.0	0.2533	-0.2159	0.0610	0.0731	0.8268	0.2430	0.3798

Table 11: Evaluation results on 2019-A test data for Random Forest model with different additional weights using predicted information types and regression scores to get importance scores.

Weight	HIAW	AAW	HIITF1	ITF1	ITA	PEEHI	PEE
0.2	-0.9905	-0.4953	0.057	0.1309	0.8518	0.0855	0.0647
0.5	-0.9905	-0.4953	0.0404	0.1296	0.8490	0.0855	0.0647

Table 12: Evaluation results on 2019-A test data for Random Forest model with different additional weights using regression scores to get importance scores.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM.

Wongyu Choi, Seung-Hyeon Jo, and Kyung-Soon Lee. 2018. Cbnu at trec 2018 incident streams track. In *TREC*.

Abu Nowshed Chy, Umme Aymun Siddiqua, and Masaki Aono. 2018. Neural networks and support vector machine based approach for classifying tweets by information types at trec 2018 incident streams task. In *TREC*.

Miguel Ángel García Cumbreras, Manuel Carlos Díaz-Galiano, Manuel García Vega, and Salud María Jiménez Zafra. 2018. Sinai at trec 2018: Experiments in incident streams. In *TREC*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Paul S Earle, Daniel C Bowden, and Michelle Guy. 2012. Twitter earthquake detection: earthquake monitoring in a social world. *Annals of Geophysics*, 54(6).

Po-Yao Huang, Junwei Liang, Jean-Baptiste Lamare, and Alexander G. Hauptmann. 2018. Multimodal filtering of social media for temporal monitoring and event analysis. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, ICMR '18, pages 450–457, New York, NY, USA. ACM.

Po-Yao Huang, Vaibhav, Xiaojun Chang, and Alexander G. Hauptmann. 2019. Improving what cross-modal retrieval models learn through object-oriented inter- and intra-modal attention networks. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, ICMR '19, pages 244–252, New York, NY, USA. ACM.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.

Shamanth Kumar, Geoffrey Barbier, Mohammad Ali Abbasi, and Huan Liu. 2011. Tweettracker: An analysis tool for humanitarian and disaster relief. In *Fifth international AAAI conference on weblogs and social media*.

Edward Loper and Steven Bird. 2002. Nltk: the natural language toolkit. *arXiv preprint cs/0205028*.

Ning Lu, Hesong Wang, and Zhen Yang. 2018. Bjut at trec 2018: Incident streams track. In *TREC*.

Richard Mccreadie, Cody Buntain, and Ian Soboroff. 2019. Trec incident streams: Finding actionable information on social media.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Taro Miyazaki, Kiminobu Makino, Yuka Takei, Hiroki Okamoto, and Jun Goto. 2018. Nhk strl at trec 2018 incident streams track. In *TREC*.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Barbara Poblete, Jheser Guzmán, Jazmine Maldonado, and Felipe Tobar. 2018. Robust detection of extreme events using twitter: worldwide earthquake monitoring. *IEEE Transactions on Multimedia*, 20(10):2551–2561.

Johan Reitan, Jørgen Faret, Björn Gambäck, and Lars Bungum. 2015. Negation scope detection for twitter sentiment analysis. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 99–108.

Konstantinos Sechidis, Grigorios Tsoumakas, and Ioannis Vlahavas. 2011. On the stratification of multi-label data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 145–158. Springer.

Hamada M. Zahera, Rricha Jalota, and Ricardo Usbeck. 2018. Dice @ trec-is 2018: Combining knowledge graphs and deep learning to identify crisis-relevant tweets. In *TREC*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.