EPIC_MR Participation in the TREC 2018 Incidence Stream Track

Simon W. Y. Chung, K. K. Lo

Info@machine-reading.com

Abstract

This paper describes our participation of the EPIC_MR group to the TREC 2018 Incident Streams Track. The target of the track is to monitor the social media and classify different type of information to help different response agencies. This paper describes our approach to use the words with Wikipedia articles to build the training vector, and also the result and comments of our runs.

Introduction

In this short paper, we describe the methods we used to build our training set, how do we evaluate the different type of classification algorithm, the result of our runs and the comments based on our current work.

The goal of TREC 2018 Incident Streams Track is to analyze social media emergencies information like requests and report, so the responding units can quickly get useful information and help in planning. In this track, tweets are used for the testing set. The track will analyze numbers of tweets which is fall under different event such as earthquakes, flood and classify each tweet into 25 high-level event type. Some examples of the event types are Request for goods, request for information, calling to action, report, and etc.

For starter, we proposed a simple way to filter words with meaning, train the classification algorithm by tagging the event type and then use them to categorize the tweets into high-level event type in a fast manner.

The paper mainly focuses on the system overview and result. System overview will talk about the data and how our system is constructed. Result will talk about our test result and discussion about them.

System Overview

Data Collection

All topic, information type and tweets data are provided by task coordinators. There are 6 topics with around 1.3k tweets for training and 15 topics with around 22k tweets for testing. The system classifies the tweets into 25 high level types.

Training

To make the dataset (the tweets) more meaningful for training, we do a couple of adjustment to the words. First, we remove punctuation and non-alphanumeric characters to create an English only training set in order to reduce complexity and improve accuracy. Second, we make all characters to normalize the word forms. Third, we filtered out the common English stop words which is provided by nltk library and some self-defined stop word such as "RT, meaning retweet" as these words are less meaningful to our system. Fourth, strings like URL are also filtered out as they are also relatively less meaningful to the message.

The remaining are the words we are interested in and relatively meaningful. However, some words are meaningless with phrases. We decided to combine those words into 2-gram and 3-gram combinations to find possible combinations. For the 2-gram and 3-gram combinations, we search those possible phrases with Wikipedia knowledge base to see if there are any matches. For those phrases with Wikipedia matches, we regard them as meaningful phrase and mark them as new words. Combining the original words and possible phrases, we use them to train as our training vector.

For the training, we used different models to test for accuracy. They include CART (Classification and Regression Trees), Gaussian Naive Bayes, Neural network Multi-layer Perceptron, Nearest Centroid, Random Forest Classifier, Gradient Boosting Classifier. All the training and fitting are run using sklearn python library.

Model Selection

With different model, we use the training dataset to evaluate its accuracy. We pick 80% of data to train and another 20% to fit. The result shows that CART (Classification and Regression Trees) and Random Forest Classifier have relatively higher and similar accuracy comparing to the others. Therefore, for the real data set, we decided to use CART and Random Forest Classifier with different number of trees to our final learning.

Result

Below is our submitted run performance.

(Multi-type)	Median	CART	RF	RF	RF
Information Type Precision (positive class, multi-type, macro)	0.1827	0.1401	0.1446	0.1625	0.1496
Information Type Recall (positive class, multi-type, macro)	0.0784	0.0674	0.0647	0.0645	0.0712
Information Type F1 (positive class, multi-type, macro)	0.0825	0.0819	0.0673	0.0672	0.0873
Information Type Accuracy (overall, multi-type, macro)	0.8993	0.8952	0.9005	0.9004	0.8964
(Any-type)					
Information Type Precision (any valid type, micro)	0.3978	0.3339	0.4415	0.4401	0.3514
Information Type Recall (any valid type, micro)	0.6165	0.6004	0.5502	0.5502	0.6064
Information Type F1 (any valid type, micro)	0.4775	0.4291	0.4899	0.4890	0.4449
Information Type Accuracy (any valid type, micro)	0.3385	0.2876	0.3533	0.3525	0.3018

We mainly focus the F1 score and Accuracy, the score is close to the Median score comparing to the others. As you can see, the score is not ideal.

One of the reasons is the training set is not good and meaningful enough. It should be better if we linking the text with more knowledge base. Also, social media text are usually not standard English, there it should be better if we can unify the text with same word such as text correction and unify to single verb tense or even part of speech. Also taking user profile into account may be also useful to improve the model. With more experiments to discover more combinations, the result should be improved.