Demonstration of Carrier Network Failure Detection by Analyzing Twitter

Masahiro Yokota, Kei Takeshita and Ken Nishimatsu

NTT Network Technology Laboratories
Midori-Cho 3-9-11, Musashino-Shi, Tokyo, Japan
Email: yokota.masahiro,takeshita.kei,nishimatsu.ken@lab.ntt.co.jp

***Abstract*—In this demo, we present a network failure detection system that is constructed by extracting tweets related to network failures and estimating the failure area. The demo is based on the work "Early Network Failure Detection System by Analyzing Twitter Data" [1].**

Silent failures that cannot be detected by network operators have recently become an important issue for network carriers. To detect these kinds of failures early, network carriers need to have a way to monitor network performance from a subscriber's perspective. The traditional way to get a subscriber's feedback is through call centers or email. However, those channels are not effective for detecting problems early or for understanding them in their entirety because subscribers generally do not call the call center until they are certain the problem is due to the network, and even then, usually only a few people will actually call. Therefore, we have studied a way to monitor a social networking service (SNS) (namely, Twitter) to discover problems affecting subscribers. For example, if we see a surge in the following kinds of tweets, we suspect a network failure, especially in data communication services. We call these kinds of tweets *network-failure tweets*

> *Why cant i send text messages?*
> *My phone won't let me make or receive calls. Is there a [Telecom Name] outage?*

The architecture of the network failure detection system is shown in Fig. 1. This system detects network-failure tweets from Twitter streaming data and then alerts the network failure area to network operators. This system consists of four components: keyword filtering, machine learning filtering, location estimation, and alerting (see [1] for more details).

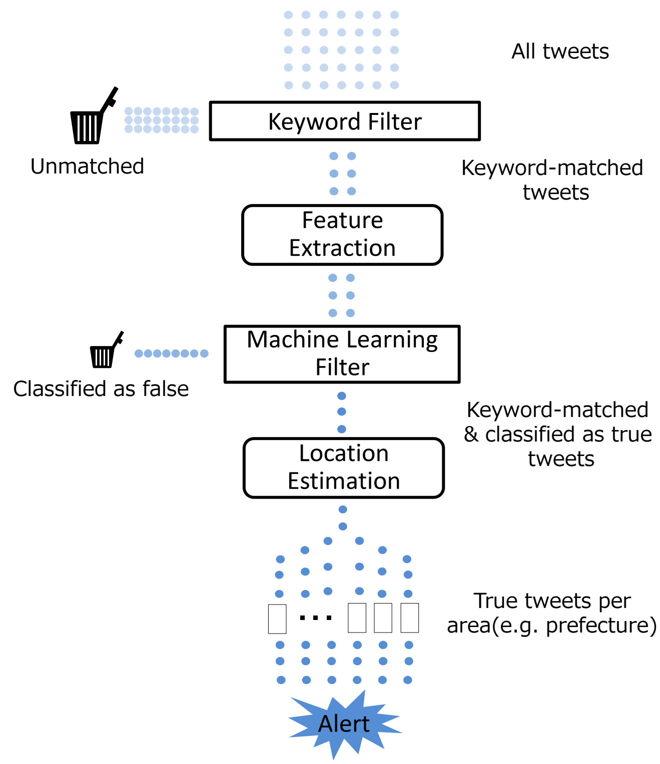

Fig. 1: Architecture of network failure detection system

Keyword Filtering:
: The keyword filter collects tweets that are possibly associated with a network failure by recognizing a wide range of keywords related to failures (e.g., failure, outage, trouble) and carriers. In this demo, we set words that appear frequently in manually classified network failure tweets as keywords about failures.

Machine Learning Filtering:
: The machine learning filter classifies each tweet into the categories of true (*network failure tweets*) and false (*false tweets*). We use supervised learning that uses the data set of training examples. Each training example consists of a pair of a tweet text and a label indicating whether or not the tweet is a *network failure tweet*. A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. In this demo, we use SVM (Support Vector Machine) for the supervised learning algorithm because it satisfies our requirements for speed and accuracy in our research. We also explain the feature extraction. The feature extraction converts raw text into values so that an algorithm can calculate the inferred function. We use bag-of-words, which is commonly used in feature extraction in natural language processing. In this demo we use the top 4,000 frequent words in feature extraction.

Location Estimation:
: This component estimates the location of users who tweeted about network failure. We use geographical names such as those for station, city,

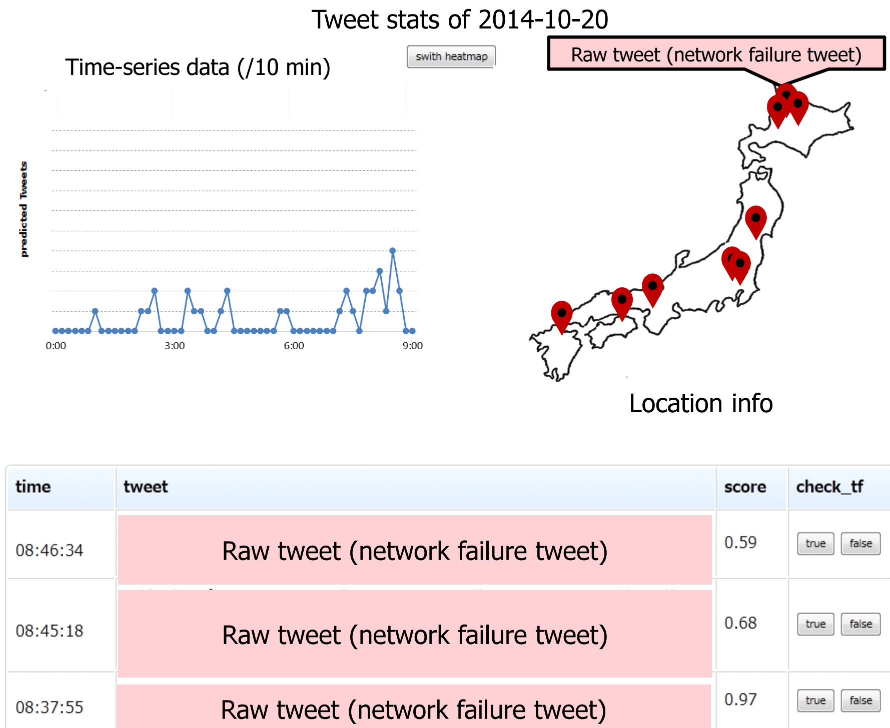

Fig. 2: Screenshot of network failure detection system

or landmark as hints to estimate the tweeter's location. Therefore, we use a gazetteer to estimate the location. We propose a method to estimate the location by applying kernel density estimation of past tweets.

Alerting:
The alerting element alerts the operators that the network may have some problems when the number of *network failure tweets* exceeds a certain threshold, which is based on the number of failure tweets occurring in the same time band on the same day in the past week. The operators check the tweets and the location information for each tweet.

First, the system collects Twitter streaming data (e.g., using Twitter Streaming API). Second, the keyword filter loosely extracts some tweets associated with failures. In this stage, the filtered tweets are a mix of *network failure tweets* and *false tweets*. Third, the machine learning filter correctly extracts network failure tweets by using SVM. Fourth, the location estimation unit estimates the location of users who tweeted about a network failure. Fifth, the alerting unit alerts the network operator if the number of network failure tweets is larger than that in the same time band on the same day in the past week. Finally, the system displays network failure information to the network operator.

Fig. 2 shows a screenshot of the system when a network failure occurred in Hokkaido prefecture. Our system shows time-series data of the number of tweets per 10 minutes (upper left graph). The network operator can adjust the graph to show data for the whole country and for individual prefectures. We also show the location information on the map (upper right image). The system plots the area of the users tweeting network-failure tweets, so areas with many plot points (e.g., Hokkaido prefecture) are network failure areas. The network operator can identify network failure areas by looking at the map. The network operator can also switch the map between a heat map and a pin map. A heat map makes it easy to understand the difference in the number of tweets. A pin map, on the other hand, makes it easy to understand the subscriber's perspective by showing the tweet text according to the area. The table lists the tweeted time, the tweeted text, and the score obtained from regression and feedback check buttons. The network operator can observe the tweet text to judge the believability of the system alert. If the network operator finds misclassified tweets, the operator can use the feedback button to improve the SVM model. For example, the operator clicks the " true " button when this model classifies false tweets as network failure tweets. This classification model is in tune with the user in this way.

In this demo, we show what this system can detect the network-failure area and tweet on a certain network-failure day.

References

[1] K. Takeshita, M. Yokota, and K. Nishimatsu, "Early network failure detection system by analyzing twitter data," in *Integrated Network Management (IM 2015), 2015 IFIP/IEEE International Symposium on*, 2015.