NiRec: Need Recommender for Hurricane Disaster Relief

Long H. Nguyen, Siyuan Jiang, Hashim Abu-gellban, Hanxiang Du, Fang Jin
Texas Tech University
Lubbock, Texas
(long.nguyen,siyuan.jiang,hashim.gellban,hanxiang.du,fang.jin)@ttu.edu

ABSTRACT

It is of paramount importance to know the situations of people who undergone disaster events and be aware of their updates, yet it is not an easy job to accomplish in the chaos of a disaster. To facilitate advanced disaster relief organization and efficient supplement distribution, we develop NiRec, a social media based need prediction prototype that predicts needs for victims across the affected area. NiRec first extracts problems and concerns posted by victims of disaster–hurricane, in our case study; then displays the statistics to offer an overview for awareness and further analysis; and last, predicts the needs such as *"diaper", "boat", "canoe" and "shelter" etc.* for disaster relief planning.

CCS CONCEPTS

• **Information systems → Spatial-temporal systems**.

KEYWORDS

Need Prediction, Multilabel classification, Concern flow

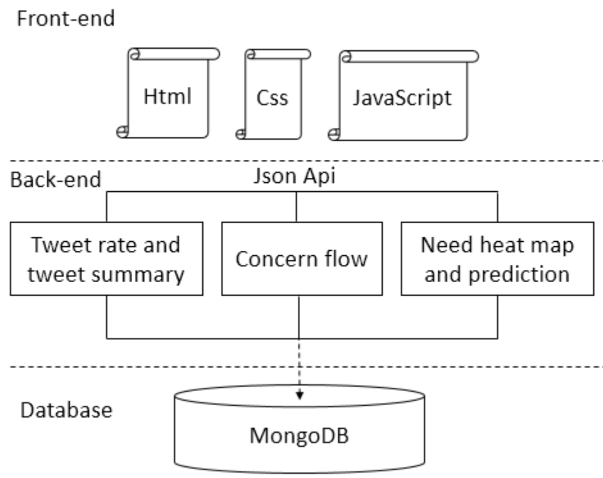

Figure 1: The overview design of the NiRec prototype.

1 INTRODUCTION

A crucial and time-sensitive task during any disaster is to distribute resources to the target groups and locations. However, this is challenging as the situations keep changing over time, not to mention the complexity of forecasting victims' needs dynamically with the respect of region characters. When Hurricane Harvey hit Houston area in 2017, numerous victims asked for transportation, medical help, and supplements. If these needs can be predicted accurately with a reasonable leading time, commodity flows can be arranged in advance, thus mitigating shortages and enhancing rescue operations. Social media has been proved to be a solid source from which reliable disaster-related information can be extracted [3]. On this account, we developed NiRec, a need prediction prototype using social media data to predict victim's needs before, during, and after disasters, aiming to facilitate timely rescue and efficient supplement distribution.

In general, NiRec provides a timely overview of the affected area by collecting the data from Twitter, displaying the data and making hourly need predictions. The updated data also helps users observe the situation and track the changes. The need prediction provides valuable guidance to relief work as it foresees people's needs accurately. Besides, NiRec makes all the information easy to understand with the help of visualization. Figure 1 shows the overview of the architecture design and the core components of the prototype. The front-end of our prototype is a web application. The back-end composes of several modules such as tweet rate, tweet summary, concern flow, heat map and need prediction. These core processing modules interact with the underlying database system. Although NiRec demonstrates the need prediction on Hurricane Harvey, it is also applicable to any other disaster events in different areas.

Most of the previous works such as [3, 6, 7, 9] mainly focus on assisting current situations via dissemination of information, optimization of volunteers and other authorities collaboration. Our proposed solution has the following advantages:

- NiRec integrates tweets posted by victims, thus is able to track the real time situations during disasters and provides a reliable overview.
- NiRec makes prompt need predictions per region, which helps clarify relief priority for different places in order to deliver victims what they needed most at the shortest time.
- NiRec also works as an archive and an analysis tool with its capacity to study the past disaster events for further reference.

SSTD '19, August 19–21, 2019, Vienna, Austria

ACM ISBN 978-1-4503-6280-1/19/08.
https://doi.org/10.1145/3340964.3340985

2 FRAMEWORK AND METHODOLOGY

Figure 3 is the landing page of NiRec. It shows needs of people across regions on the map of Houston. It also supports other types of statistics for analysis such as tweet rate, word cloud, concern flow and need related prediction per region to enhance the usability of the application. The prediction mechanism is based on multilabel

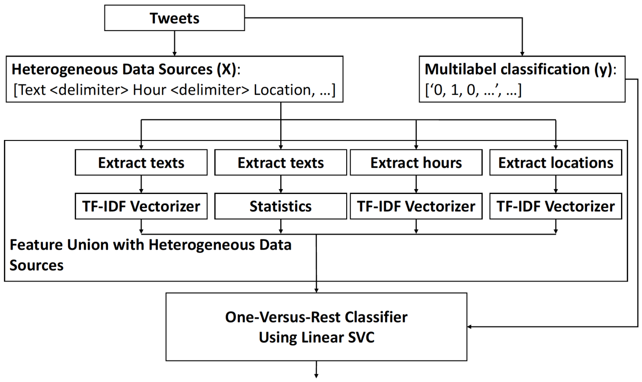

Figure 2: The proposed algorithm. The data flow starts with raw tweets (top) to process data into X and Y. The feature union (middle) is to combine the vectorization and statistics features. The bottom is to build the classifier for prediction.

classification algorithm. Detail of the algorithm is described in the following sections:

2.1 Dataset and Preprocessing

We crawled tweets with Harvey hashtag during Hurricane Harvey from August 23 to September 01, 2017. The tweets contained spatial information from all towns of Houston. The total number was 15, 379. All stopwords were removed to improve the performance of the prediction model. For instance, *"in", "am", "any", "a", "an"*, and *"the"* were deleted using NLTK Python Library [2]. In addition, we removed non-English examples as well as non-ascii characters[4, 8]. Furthermore, words were also normalized (by doing lemmatization or converting synonyms to a common word) to increase their importance weight. To build the prediction model, the dataset is split by 70% for training and the rest (30%) for testing purpose.

2.2 Need Labeling

We use top frequent words and word co-occurrence matrix to support our manual need labeling process. First, top 100 frequent words within one hour block of each region are extracted and play as candidates for need extraction. Later, a co-occurrence matrix of these words with other words in the tweets is setup to understand the context of those words in the tweets. By looking at the frequency of each pair of words, we can learn and understand those frequent words better and be able to identify whether a candidate word is an actual need or not. If there is a confusion in deciding needs, the resolution is based on the number of common agreements between three graduate students who do the labeling task. This step will produce a training set to train the need prediction algorithm before deployment.

2.3 Need Prediction

Our purpose is to apply a fast classification algorithm which is suitable for the multilabel problem. For this endeavor, we employed the One-Versus-Rest multiclass classification method for the Hurricane examples, in order to predict the needs. The method requires applying the feature union algorithm before invoking the classification algorithm. The accuracy, precision, recall and F1-score metrics of our model are 0.81, 0.72, 0.81 and 0.75, respectively. These results show how efficient our proposal algorithm to anticipate needs for unseen test examples.

2.3.1 *Feature Extraction and Vectorization.*

We packed three attributes, hour, location, and text for every tweet into a single element. We passed all packed tweets to the feature union algorithm [1, 5] . For all elements, attributes were extracted and the TF-IDF (i.e., Term Frequency - Inverse Document Frequency) vectorization features were calculated as well as a customized statistical feature (i.e., the length of a tweet's text and the number of dots in it) was computed. After that, all features were combined with their weights as shown in Figure 2. We employed two processes (i.e., extract text), as we ran the feature union in parallel using asynchronous pipeline .

To process text data faster, we employed a common text vectorization algorithm called TF-IDF, to calculate the word frequency vector.

$$tf(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \tag{1}$$

$$idf(t,D) = \log \frac{N}{1 + |\{d \in D : t \in d\}|} \tag{2}$$

$$tfidf(t,d,D) = tf(t,d) \,.\, idf(t,D) \tag{3}$$

where *f(t, d)* denotes the number of times the word *t* in document *d*, and *D* denotes all documents. In our method, a document is a tweet's text, hour, or region name. *D* is a corpus with size *N*. In formula 2, one is added to the divisor to prevent divide-by-zero when *t* does exist in *d*. We passed the transformer weights of the feature union having 90% for tweet text feature, 1% for tweet text statistics feature, 25% for region name feature, 25% for hour feature. We tried several settings for the transformer weights taking into consideration the effectiveness of the features. We found that the vectorization of tweet texts was more effective than the lengths of the texts (e.g., same need might be in variant lengths of tweets) and other features which were handled as string data types and were helpful to determine the need in some special cases.

2.3.2 *One-Versus-Rest Classification.*

The unified features (X) calculated in Section 2.3.1 and the multiclass label vectors (y) of the tweets were sent to the One-Versus-Rest classifier *H* (i.e., One-Versus-All) as shown in Figure 2.

One-Versus-Rest is applied to perform classification of K class labels by simplifying the K labels problem into K binary label problem. This reduction is useful since the classification algorithms (e.g., SVM, Linear Support Vector Classification "Linear SVC") classify data set with one label (i.e., only one class label value of 2 or more classes) for each instance. This method trains the data using the k^{th} classifier for the model h^k (x), by using ones (i.e., positive output) if the data records are classified as k^{th} class label, and zeros (i.e., negative output) otherwise. It repeats this step for all class labels, to generate K models, $h^k(x) = P(y = k|_x)$ where k = 1 to K.

The method evaluates data instances (x) against all models, to predict (y); the prediction for an instance is k class label if the output of h^k (x) is positive; for non-unique output, tie-breaking techniques are used. This evaluation process is repeated for all K-1 models (i.e., h^i (x) $|_{i = 1, ..., k-1, k+1, ..., K}$).

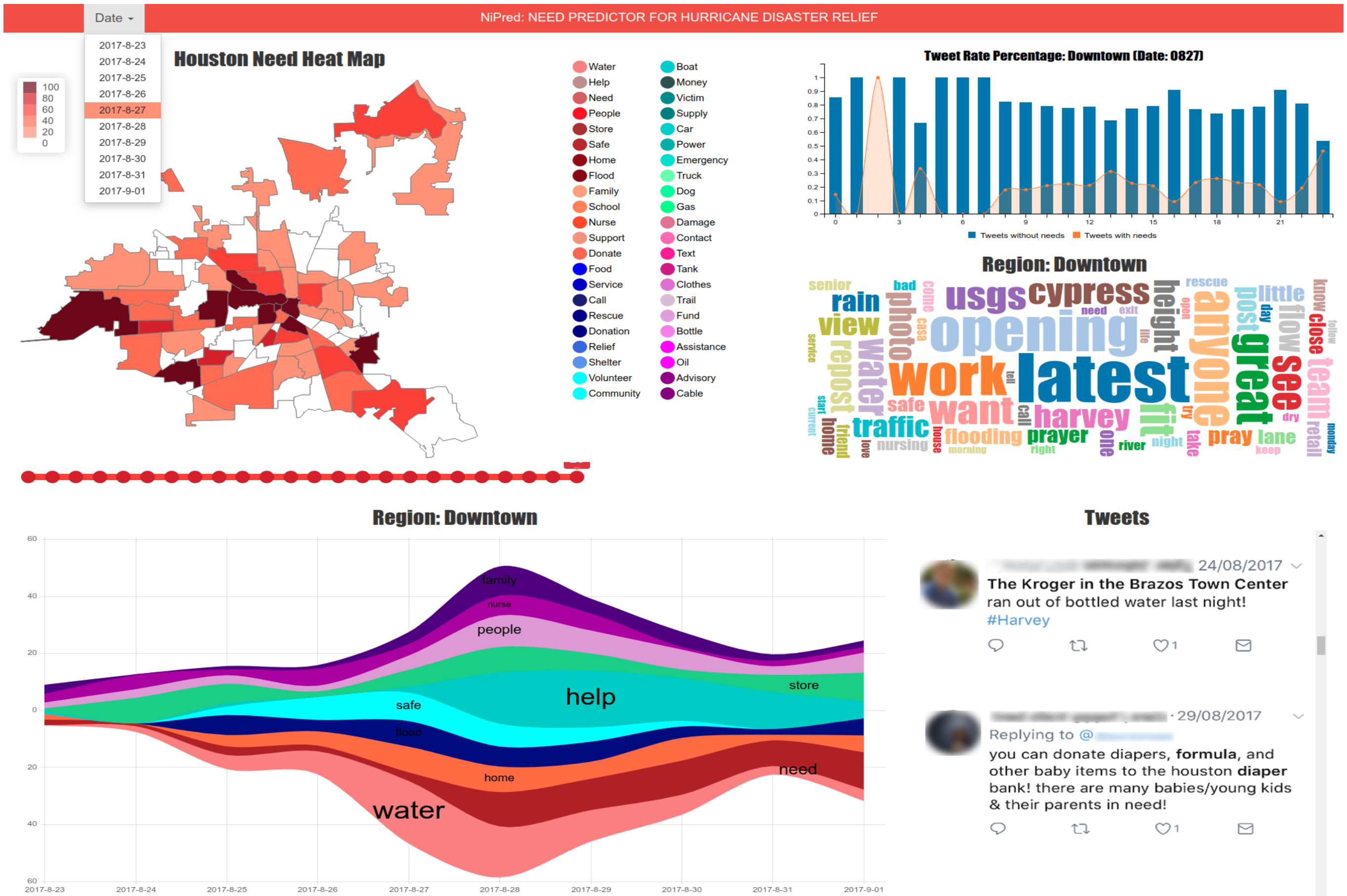

Figure 3: The NiRec landing page demonstration. Top left displays a heat need map of Houston. Top right is a tweet rate and a word cloud of Houston downtown where bottom left and right present concerns of people over time and tweets details.

$$ar\partial\ max_k\ h^k(x) \tag{4}$$

We used 5-fold cross-validation method. The training data were split randomly into 5 groups, to train the model using 4 folds and evaluate them using 1 fold. This process was repeated 5 times such that an instance can be evaluated once. Cross-validation is useful to estimate the strength of the model against an unseen dataset. The Linear Support Vector Classification (Linear SVC) algorithm was employed in our implementation due to its scalability and efficiency to convergence, which allows quick prediction that satisfies crisis time.

3 PLATFORM DEMONSTRATION

Figure 3 is the landing page of NiRec. The main three features of this platform are:

- *Tweet rate and tweet summary*: The feature shows the ratio of tweets with and without needs based on their publishing frequency. Also, it presents the tweet content summary per region. Based on this, it gives a summary to the end user about the people's concerns via the word cloud. Moreover, it provides an overall evaluation of the importance of needs via the frequency of tweets.
- *Concern flow*: This feature that is shown in Figure 5 helps users understand the crisis and changes of concerns over time. It supports understanding the sequence of the problems raised by people. Also, it provides tracking of the time period during the occurrence of the issues.
- *The need heat map and prediction*: The map is based on our proposed algorithm to predict the needs for relief planning, described in Section 3.3. The prediction is for each region. The darker color represents the more needs in that area.

Overall, this is a prototype for one of the first tools featuring need prediction by regions for Hurricane relief planning within a big city. Details of the supporting features are discussed in the following sections.

3.1 Need Tweet Rate and Tweet Summary

The top right of Figure 3 is the hourly tendency of publishing tweets, between need-related tweets and non-related tweets in one day. On August 27^{th}, the number of published tweets with needs in