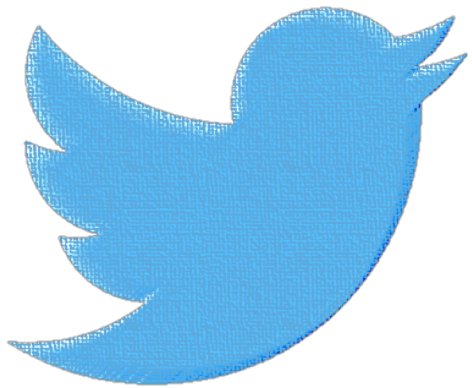
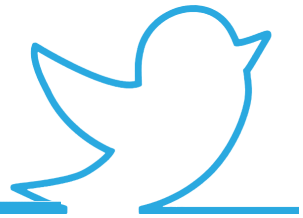
Developing a Twitter-based traffic event detection model using deep learning architectures

Big Data Mining Lab

Youngjin Kim



Contents



- Introduction
- Twitter-based TED model
- Data collection
- Methodology
  - Word embedding
  - Convolutional neural network
  - Recurrent neural network
- Experiment

Introduction

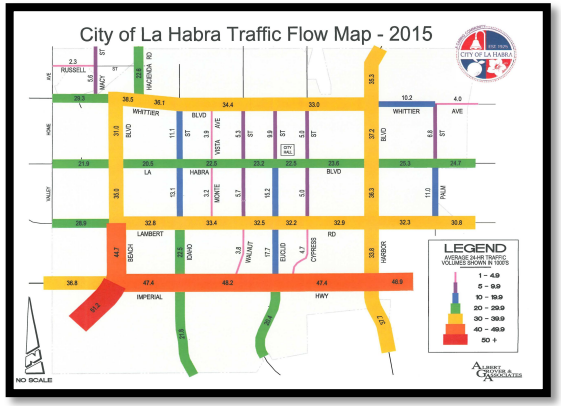
- Traffic control and management strategies
  - Predictable conditions
  - Unpredictable conditions

PREDICTABLE

UNPREDICTABLE

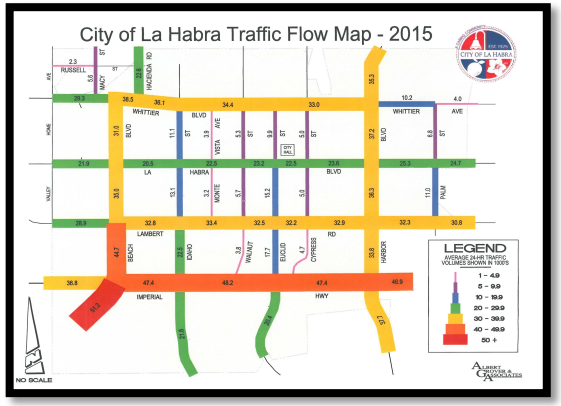
Introduction

- Traditional Traffic flow sensors
  - CCTV cameras
  - GPS
  - Analysis of collected data
  - Radio report

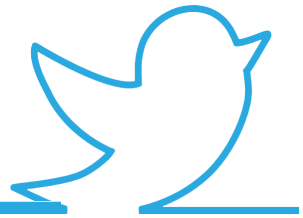


Introduction

- Traditional Traffic flow sensors
  - Unacceptable false-alarm rate
  - Long detection time

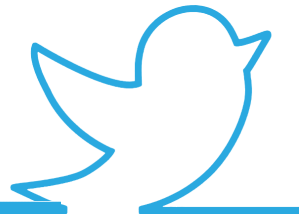


Introduction



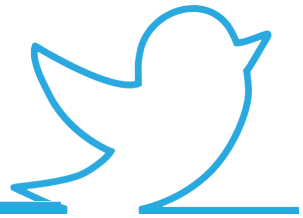
- Growth of social networking tools
  - Rich information
  - 330 million twitter users
  - Real-time reporting

Introduction



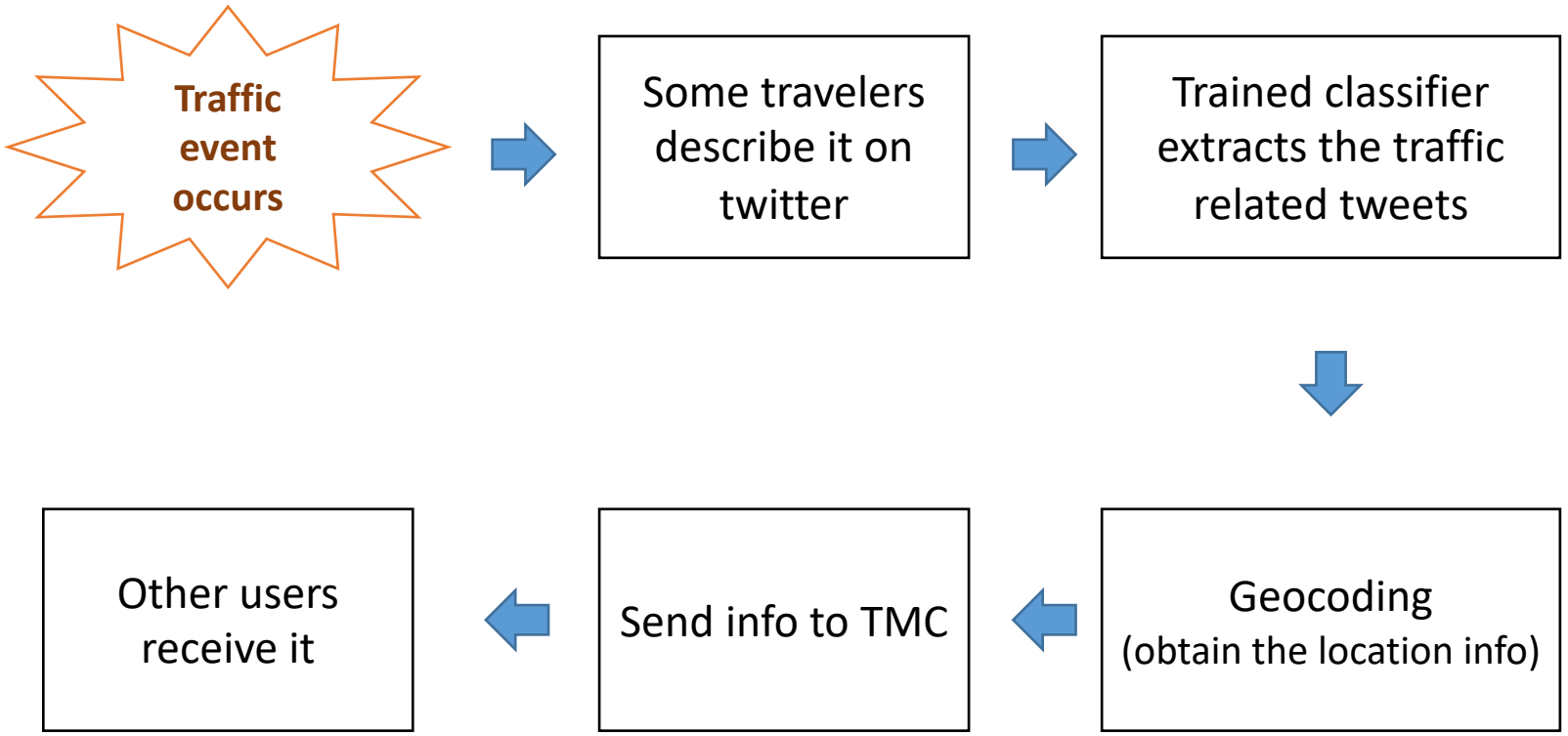
- Twitter for traffic monitoring system
  - No sparse coverage
  - Efficient & ubiquitous
  - Inexpensive & widespread
  - People as dynamic social sensors

Twitter-based TED model

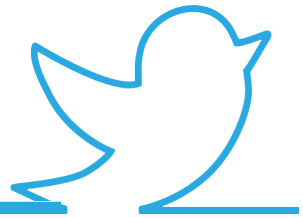


- Collecting and labeling a large volume of tweets
- Developing a **traffic event detection model** using deep-learning architectures
  - Get feature vectors from word-embedding tools
  - Discriminating traffic-related from non-traffic tweets by using CNN and RNN

Twitter-based TED model



Data collection



- 51,000 tweets using the twitter APIs
- Labeled them into three classes
  1. Non-Traffic (NT)
  2. Traffic Incident (TI)
  3. Traffic Conditions and Information (TCI)
- Internal labeling: labelers are the authors

Data collection

- Traffic-related dictionary
  - Most frequent words
  - Belong to emergency service providers
  - Make tokens from their tweets

@511northernva
@511Gerogia
@511NYC
@511nyNJ
…

Traffic, blocked, lane, construction, crash, delays

Data collection

	Class	Account feature
1st Dataset	Random user tweets	Random users
2nd Dataset	Traffic-related tweets	Travel information accounts
3rd Dataset	Non-related traffic tweets	Never post tweets about traffic

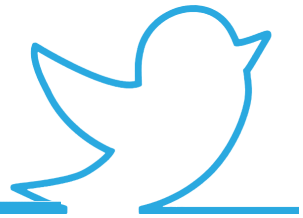
1st – Making a balance between traffic-related and non-traffic related, Slow labeling

Data collection

Dataset\Class	Traffic-related (TR)			
	NT	TI	TCI	Total
1st Dataset	8,138	5,414	2,651	16,203
2nd Dataset	377	12,022	5,462	17,861
3rd Dataset	18,259	2	0	18,261
Total	26,774	17,438	8,113	52,325
Unified dataset	25,550	17,437	8,113	51,100

- **2-class**: traffic-related (TI, TCI) + non-related (NT)
- **3-class**: NT, TI, TCI

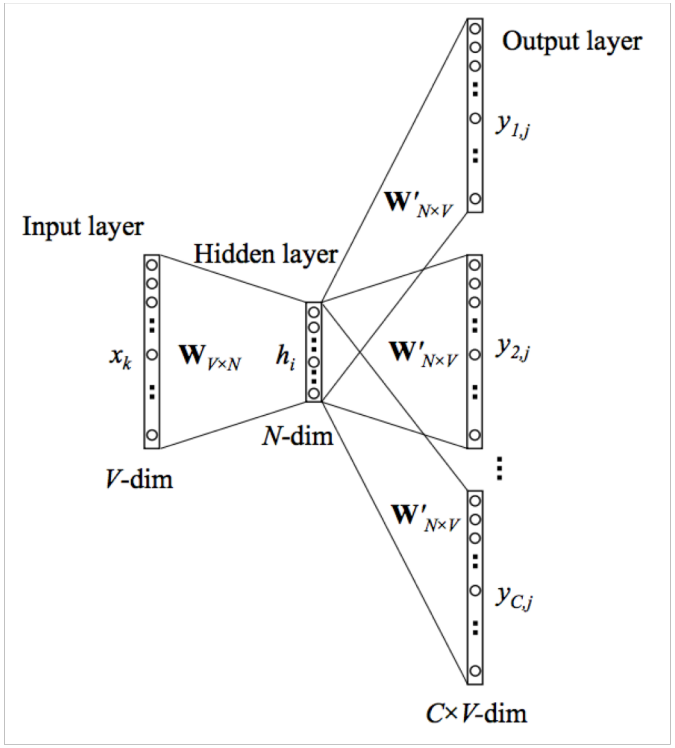
Methodology



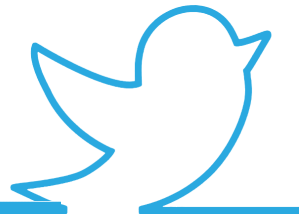
- Word embedding
  - Word2vec model
  - FastText model
  - Creating embedding layer
- Convolutional neural networks (CNN)
- Recurrent neural networks (RNN)

Word embedding

- Word2vec
  - Unsupervised learning algorithm
  - one hidden layer is trained
  - V: the number of words in the vocabulary
  - N: dimensionality of the word vectors
  - W: weight matrix ($V \times N$)

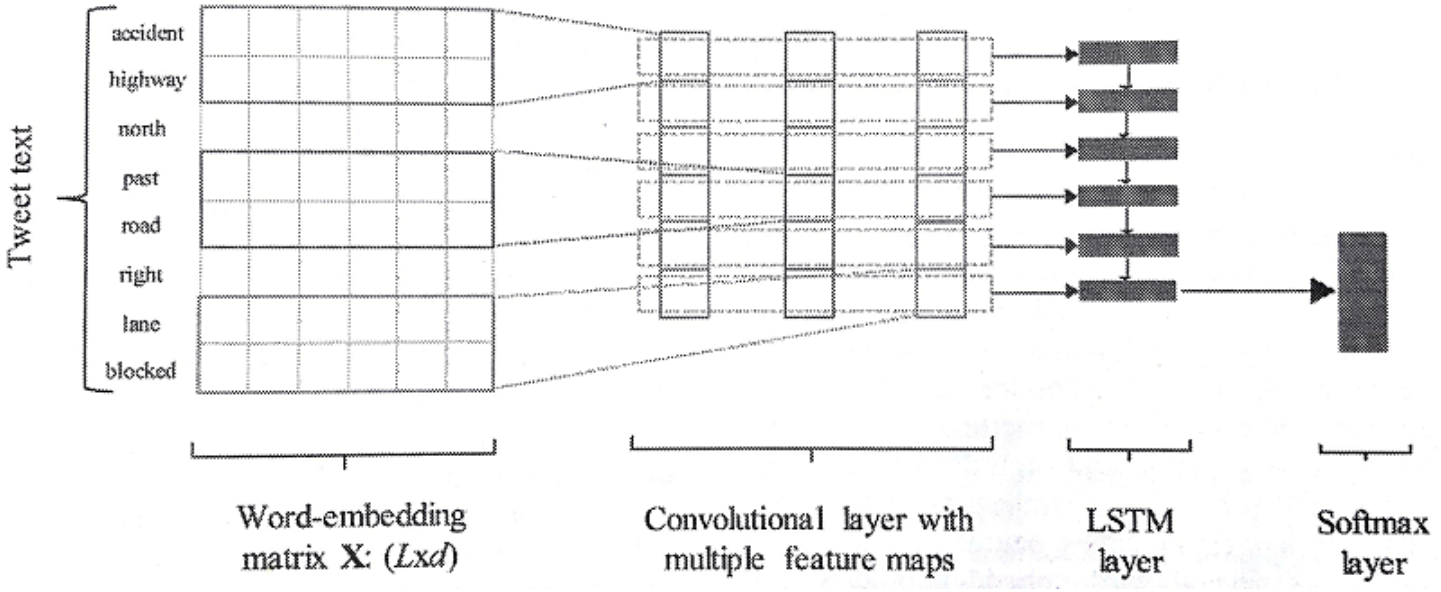


Word embedding



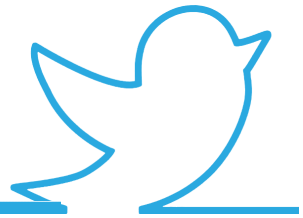
- Every tweet is mapped into a fixed-sized feature matrix $\mathbf{X}$
- Each matrix need to have the same size for both RNN & CNN
- $L(\text{maximum sequence length})$ should be defined
  - More than L – truncated to L size
  - Less than L – padded with zero values

Word embedding



Deep-learning architectures for tweet classification

Convolutional neural networks



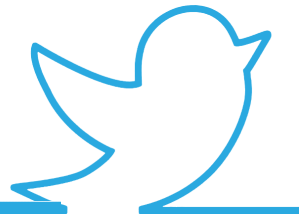
- Convolutional layer
  - filter $\boldsymbol{F}$ has size $(n \times d)$
  - n: number of consecutive words (i.e., n-grams)
  - d: width of filter equal to the word-vector dimension
- feature map

$$\boldsymbol{c} = [c_1, \ldots, c_i, \ldots, c_{L-n+1}]^T$$

$$c_i = f(\boldsymbol{F} \cdot \boldsymbol{X}_{i:i+n-1} + b)$$

  - b: bias term
  - f: non-linear activation function

Convolutional neural networks



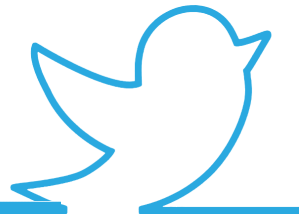
- New world-vector representation matrix

$$\mathbf{X}_{new}: [L - n + 1, K]$$

  - K: same size as $\boldsymbol{F}$, filters that from ReLU activation function

- The i-th row of the matrix $\mathbf{X}_{new}$ is a new feature representation of the n-grams in the original matrix $\mathbf{X}$

Recurrent neural networks



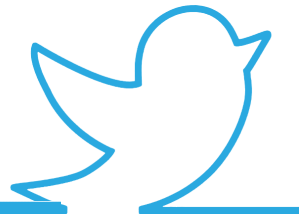
- Using LSTM
- Input layer for the LSTM layer is the matrix $\mathbf{X}_{new}$ as obtained through the CNN layer
- Last step of the LSTM layer is fed into the $softmax$ layer
- The $softmax$ function in the last layer classifies tweets

Experiment

- Accuracy – fraction of tweets in the test set are correctly classified
- F-measure – F-measure for ever tweet class

$$F - meaure = \frac{2 \times Precision \times Recall}{(Precision + Recall)}$$

Experiment

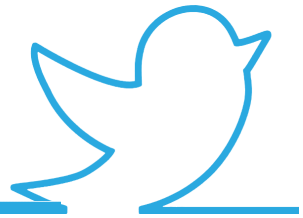


Model	word2vec				FastText				Random			
	2-class		3-class		2-class		3-class		2-class		3-class	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Only CNN	**0.986**	**0.986**	**0.974**	**0.974**	0.986	0.986	0.973	0.973	0.501	0.398	0.493	0.370
Only LSTM	0.984	0.984	0.972	0.972	0.985	0.985	0.970	0.970	0.503	0.502	0.500	0.336
CNN+LSTM	0.985	0.985	0.972	0.972	0.986	0.986	0.971	0.971	0.499	0.498	0.500	0.334

Experiment

CNN-based for 2-class		Predicted class		
		NT	TI & TCI	Recall%
Actual class	NT	4,999	111	97.8
	TI & TCI	46	5,065	99.1
	Precision%	99.1	97.9	

CNN-based for 3-class		Predicted class			
		NT	TI	TCI	Recall%
Actual class	NT	5,031	57	32	98.3
	TI	45	3,411	40	97.6
	TCI	34	73	1,513	93.4
	Precision%	98.5	96.3	95.5	



Thank you