NATURAL LANGUAGE PROCESSING WITH DISASTER TWEETS

JIAQI LIU AND GANG LI

Abstract. Twitter has become an important communication channel in times of emergency. The ubiquitousness of smartphones enables people to announce an emergency they're observing in real-time. Because of this, more agencies are interested in programatically monitoring Twitter (i.e. disaster relief organizations and news agencies). But, it's not always clear whether a person's words are actually announcing a disaster. Out task is to decide whether a tweet is related to a disaster or not.

Contents

1. Data Process 2
2. Word Emedding 2
3. TextCNN 3
4. Model Training 3
5. Result 4

Date: 2021-07-23.
Key words and phrases. Natural Language Processing, TextCNN, ...

1. Data Process

- *Word Spliting*
  In online media, texts may contain various seperations. The vital task of spliting words by regex matching is to list all the possible seperations. And some specific forms, such as abbreviation, deformation and links need to be matched separately and primarily.
- *Sentence Padding*
  Pad all sentences to max length 209 to ensure they can be processed equally by the CNN net.
- *Word Encoding*
  Encode the words mentioned by indexing them in the dictionary. Thus, the following embedding operations can be done by finding the indexed vectors.

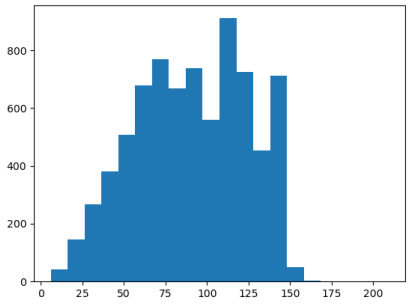


Figure 1. Sentence Lengths

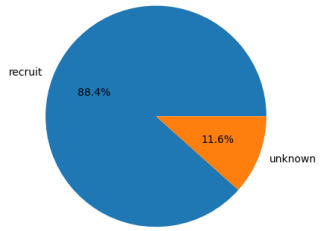


Figure 2. Recruit Proportion

2. Word Emedding

- *GloVe*
  Word vector pretrained by *Stanford*.

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

- Based on the co-occurence matrix and represent the correlation of words by calculating the ratio.

Ratio	word j,k related	word j,k non-related
word i,k related	close to 1	very big
word i,k non-related	very small	close to 1

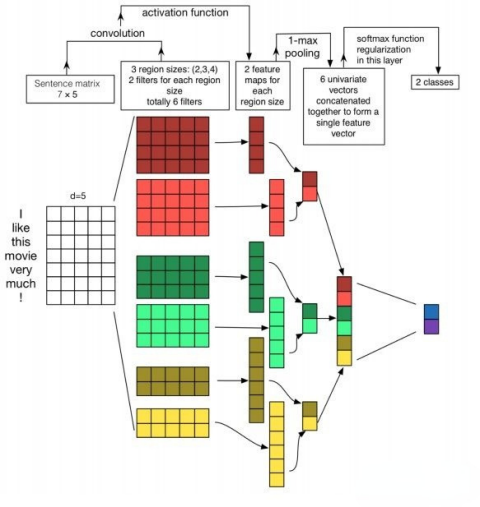
- Construct the target function which:

$$F(w_i, w_j, w_k) = \frac{P_{ik}}{P_{jk}}$$

Both sides of the equation represents correlation between words.

3. TextCNN

- Suppose we view an 100 word vector as signals on 100 channels. We can extract text features by using 1D convolution kernal over the channels.



4. Model Training

Training Parameters are listed as followed:

- $batch\_size = 10$
- $learning\_rate = 0.005$
- $loss = BinaryCrossEntropy$
- $optimizer = AdamOptimizer$
- $accuracy = (\sum_{i=0}^{N} 1 - |\frac{1}{2}(sign(\hat{y}_i - 0.5) + 1) - y_i|)/N$

Accuracy & Loss during the training process is shown in Figure 3 and Figure 4:

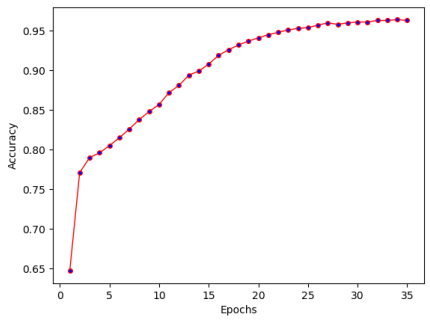


FIGURE 3. Training Accuracy

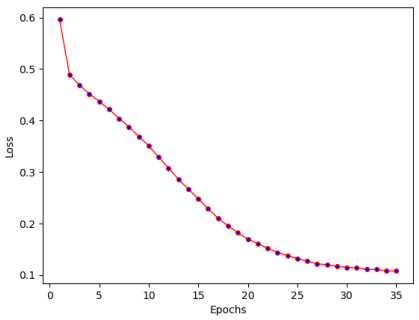


FIGURE 4. Training Loss

- An interesting phonomenon happened near $epochs = 27$, as figure 5 shows.

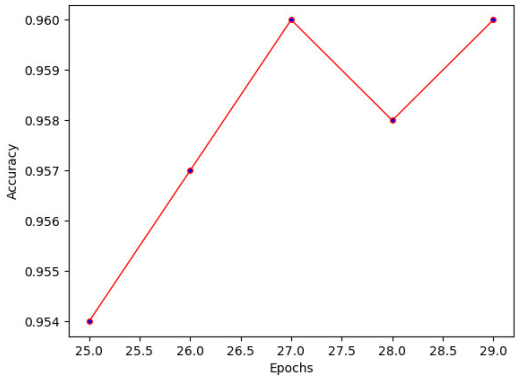


FIGURE 5. Local Accuracy

- By testing epochs near 27, we find the best $epochs = 27$

5. RESULT

- Final accuracy reahces 0.78455 with 27 epochs.
- Rank:2349/3625

(A. 1) Department of Computer Science and Technology,, Jilin University, CHangchun, China

Email address, A. 1: jqliu@tulip.academy

(A. 2) School of Information Technology, Deakin University, Geelong, VIC 3216, Australia

Email address, A. 2: gang.li@deakin.edu.au