Natural Language Processing with Disaster Tweets

Marilin Ahvenainen, Belinda Lepmets

Introduction

As Twitter has become an important communication channel in times of emergency, then different agencies (i.e. disaster relief organizations and news agencies) are interested in programmatically monitoring Twitter. Unfortunately it's not always clear whether a person's words are actually announcing a disaster.

Importance/Motivation

To be able to monitor emergencies via Twitter more efficiently using a machine learning model that is able to predict which Tweets are about real disasters and which one's aren't.

Data

We used 2 csv format datasets published in Kaggle, both together contain almost 11000 tweets.

Training data: 7613 tweets

Test data: 3263 tweets

Exemple of 'train' dataset as a table:

id	keyword	location	text	target
1	NaN	NaN	Our Deeds are the Reason of this #earthquake M...	1
4	NaN	NaN	Forest fire near La Ronge Sask. Canada	1
5	NaN	NaN	All residents asked to 'shelter in place' are ...	1
6	NaN	NaN	13,000 people receive #wildfires evacuation or...	1
7	NaN	NaN	Just got sent this photo from Ruby #Alaska as ...	1

('Test' dataset is the same, only without 'target' column)

Cleaning the data:

- removed emojis using Python demojize
- removed links and hash characters

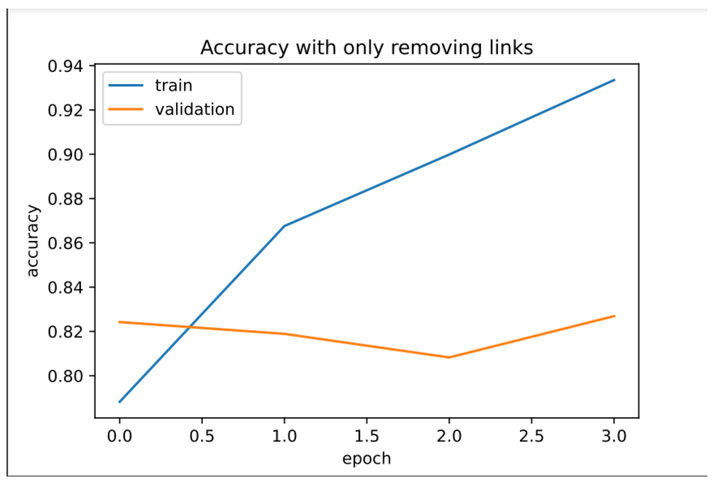
Training the model

Classification:

We performed binary classification using the pre-trained Bert model (Transfer Learning).

We received confirmation that with the Bert model, cleaning might not always improve accurency.

*best results were with only removing links



Training :

- Epochs (data iterations) : 4

Results

With the results, we achieved our goal of getting the model score over 80% (0.80).

	score
Without cleaning	0.81887
With removing emojis, links and hash characters	0.77627
Only removing links	0.82408

Best score was with only removing links and the worst score was with removing emojis, links and hash characters.

Indicators of the best result:

	precision	recall	f1-score	support
0	0.82	0.98	0.86	443
1	0.84	0.72	0.77	308

* '0' is for no disaster and '1' is for disaster

Conclusion

Even though Twitter data in general requires a lot of cleaning, then BERT model can handle the data relatively well without cleaning. With pretrained models over-fitting might become an issue. Alternatively one could try another word-embedding technique such as Word2Vec or GloVe to see if performance could be increased.