Udacity Machine Learning Engineer Nanodegree

Capstone Proposal

Gary Chen
Oct 26, 2021

Domain Background

Twitter has become an immediate source for news and emergency. With an influx of information every second, important events are announced real time on Twitter before it is broadcasted on news and televisions. One application of this phenomenon is that disaster relief organizations and polices can programmatically monitor Twitter to detect emergencies in order to respond in a timely fashion.

Problem Statement

The goal of this project is to correctly identify a disaster based on a person's word. For example, a person tweeted "The sunset tonight is ablaze!". This tweet does not indicate a disaster, as the word "ablaze" is used in a metaphorically way. The is clear to a human right away, but it is less clear to a machine.

Datasets and Inputs

The dataset will come from the Kaggle competition: https://www.kaggle.com/c/nlp-getting-started/overview

The training datasets will contain the following information:

- id: A unique identifier for each tweet
- text: The text of the tweet
- location: The location the tweet was sent from
- keyword: A particular keyword from the tweet
- target: A binary indicator, where 1 indicates a real disaster and 0 otherwise

Solution Statement

We will build a binary classification model that takes the text as an input, and predict the probability of whether one text is pointing to a real disaster or not. We will then evaluate how the model predicts against the actual answer using a standard metric.

Benchmark Model

The benchmark model we will compare with will be a basic logistic regression.

Evaluation Metrics

The evaluation metrics will be F1 score between predicted and actual answers.

The formula for F1 is below:

$$F_1 = \frac{2}{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}} = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} = \frac{\mathrm{tp}}{\mathrm{tp} + \frac{1}{2}(\mathrm{fp} + \mathrm{fn})}.$$

Project Design

Data preprocessing and feature extraction

We will extract the text features by applying the below four steps:

- Text cleaning: Common text cleaning procedures like Lemmatization, Stemming, and Stop Words Removal
- Tokenization: Tokenizing all sentences
- Creating vocabulary and Transforming text: We will create a vocabulary of the tokens. Then, the Tokenized sentences will be mapped to unique ids.
- Word embedding: Generate word vectors by using Global Vectors for Word Representation (GloVe)

Splitting the data

The training data will be split into 75% training sample and 25% validation sample.

Model Training

We will apply LSTM (Long short-term memory) and AutoGluon to predict whether one text indicates a real disaster (1) else (0).

- LSTM is preferred over normal FFNN as it captures the sequence on inputs (in this case, text sequences)
- AutoGluon is a AWS automated machine learning API that trains the popular algorithms like Random Forest/GBM/K-nearest neighbors/FFNN with novel techniques like multi-layer stack ensembling to boost the model accuracy.

Model Deployment

The model will be deployed and trained in AWS SageMaker if available. After the model is trained, we will deploy the model on Endpoint for real-time inference.

Kaggle submission

The final inference score on the test dataset will be submitted back to Kaggle for evaluation and scoring.

References

- Kaggle competition: https://www.kaggle.com/c/nlp-getting-started/overview
- Glove: https://nlp.stanford.edu/projects/glove/
- AutoGluon: https://auto.gluon.ai/stable/index.html