Machine Learning Engineer Nanodegree

Capstone Proposal

Ahmed Eid
(ahmedmohamedeid98@gmail.com)
November 19th, 2019

Toxic Comment Detection in Online Discussions

Proposal

Domain Background

Neutral Language processing(NLP) is one of the most important field in machine learning that deal with the interaction between the computers and humans using neutral language, in particular how to program computers to process and analyze large amounts of neutral language data. NLP plays a major role in our daily computer interactions, Machine learning and statistical models are being applied in the area of Natural Language to make life easier and better for wider sections of the society. we see it's powered software every day in our life, for example: personal assistants(Google assistant, Sari, and Cortana), Auto complete in search engine(e.g: google, Bing), machine translate(e.g: google translate), spell checking(e.g: MS-word), music generation, sentimental classification, DNA sequence analysis, speech recognition, video activity recognition and so on, Natural Language Processing algorithms are being applied to arrive at probabilistic as well as if-then-else kind of decisions.

Problem Statement

today we are in the age of social media and online news platform, every body use it, discussions comment sections are an essential space to express opinions and discuss political topics. Posting comments in online discussions has become an important way to exercise one's right to freedom of expression in the web. This essential right is however under attack: malicious users hinder otherwise respectful discussions with their toxic comments. A toxic comment is defined as a rude, dis-respectful, or unreasonable comment that is likely to make other users leave a discussion. the task of sentiment analysis is toxic comment classification.

Datasets and Inputs

Data source: Our data is taken from ***Kaggle*** competition *Jigsaw Unintended Bias in Toxicity Classification*
Data link: https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification/data
file description:

- train.csv – the training set, which includes toxicity labels and subgroups(816MB).
- test.csv – the test set, which does not includes toxicity labels or subgroups(29MB).

Data Description:

the data supplied for this competition, the text of the individual comment is found in the ***comment-text*** column. Each comment in Train has a toxicity label (***target***), and models should predict the ***target*** toxicity for the Test data. This attribute (and all others) are fractional values which represent the fraction of human raters who believed the attribute applied to the given comment. For evaluation, test set examples with ***target >= 0.5*** will be considered to be in the positive class (***toxic***).
features columns: "comment_text"
target columns: "target"/ "severe_toxicity", "obscene", "threat", "insult", "identity_attack", and "sexual_explicit".(for research)

Identity columns: "male", "female", "homosexual_gay_or_lesbian", "christian", "jewish", "muslim", "black", "white",and "psychiatric_or_mental_illness".
metadata columns: "toxicity_annotator_count", "identity_annotator_count", "created_date", "publication_id", "parent_id", "article_id", "rating", "funny", "wow", "sad", "likes", and "disagree".
Note:(I will use "comment_text" as features, "target" as label)

Solution Statement

I'm using deep learning algorithms "LSTM" which is updated version from RNN to build the model, I will do this by using tools like tenserflow/keras. Neural Networks are designed to learn from numerical data, so we will convert "**comment-text**" to matrix of numbers by using "*Word Embedding*", and In order to embedding word with some relative meanings between features, I will use pr-train embedding model "*Glove*".

benchmark model

Kaggle composition's Private Leaderboard score ("0.947") will be used as a benchmark model.

evaluation metrics

I will use "accuracy" matrix to evaluate this model and loss_function "binary_crossentropy" and minimize loss_function using "Adam" optimizer.

project design

the general sequence of steps are as follows:

- Data Exploration
  - data shape, NAN-values in data, number of samples in each class
- Data Preprocessing
  - my assumption was that most of data was labeled using ML, and only records with valid identities where better supervised by humans. Training with only valid identities records produced better performance than using random records.
  - drop NAN value and unneeded columns(e.g. metadata); 405,130 samples remaining
  - get all sample which is belong to positive_class(target = 1); 46035 samples founded
  - get equivalent number of samples from negative_class(target = 0) to positive_class
  - get last 100,000 samples to evaluate
  - making word embedding
- Model Training
  - build model with keras
  - train model on training data
- Model Evaluated
  - get accuracy of model's preforming using testing data

References

- https://www.researchgate.net/publication/315473313_What_is_hate_speech_Part_1_The_Myth_of_Hate
- https://towardsdatascience.com/why-do-we-use-embeddings-in-nlp-2f20e1b632d2
- https://machinelearningmastery.com/use-word-embedding-layers-deep-learning-keras/
- https://skymind.ai/wiki/lstm
- https://www.kaggle.com/tags/rnn
- https://machinelearningmastery.com/prepare-text-data-deep-learning-keras/
- https://github.com/keras-team/keras-preprocessing/blob/master/keras_preprocessing/text.py#L267
- http://keras.io/preprocessing/text/