M7 Project

Kaggle: Natural Language Processing with Disaster Tweets

Johnathan Wang, Nicholas Charles Vitellaro, Makenna Shae Owens, and Marc Mahanna
COSC523: Artificial Intelligence, November 15, 2023

Abstract - **This project focuses on using machine learning (ML) and natural language processing (NLP) approaches to implement models predicting which Tweet validity in disaster situations. The format of this project is that of a Kaggle competition** [1]. **Goals of this project are to 1) implement a baseline score and 2) try to improve on the baseline score.**

DESIGN CHOICES AND IMPLEMENTATION

The approach selected was to implement a baseline and an improved model intended to improve score beyond the baseline. DistilBERT is the default implementation and is provided in the Kaggle competition tutorial [2]. DistilBERT is the improved model providing the best performance after much team experimentation. For each experiment and subsequent baseline and improved implementation, data was loaded, data was cleaned, features were explored and selected, models were trained, and test results were produced.

The baseline model followed the Kaggle tutorial. Adaptation was required to get the baseline to execute without errors (details covered in subsequent challenges and obstacles section). Data cleaning and preparation was minimal. the selected feature is 'text'. Train/test split is 80%/20%. Batch size is 32; Epoch quantity is 2. Resulting performance when submitted to Kaggle was 83.450%.

The improved model was the result of much team experimentation. The first set of team experimentation performed centered around DistilBERT preprocessor and model hyperparameters. Sequence length outside of 160 characters slightly reduced performance versus baseline (+/- 0.2%). Increasing batch size resulted in slightly reduced performance as well. Increasing quantity of epochs also resulted in a slight drop in performance versus baseline (0.4%). Exploration of the optimizer size resulted in a large shift in reduced performance (20-26% lower). The second set of team experimentation centered around training dataset feature modifications. DistilBERT model and preprocessor perform roles of traditional NLP processes such as Part of Speech tagging, Tokenizer, etc… The team experimented with adding aggressive Stop Word Removal sourced from a team member's COSC-524 Regex project [3]. This was added prior to preprocessor and model fit and resulted in a 2% drop in performance versus baseline. Combined with an increase of epoch count to 3 epochs resulted in an additional 0.2% performance increase vs 2 epochs. The third set of team experimentation centered around alternative models versus DistilBERT. The team experimented with a Logistic Regression approach which achieved performance around 77% on Kaggle. Resulting best improved performance when submitted to Kaggle was 83.297% using DistilBERT for both the sequence lengths of 80 and 240 characters, all other parameters matching baseline, and no additional stop word removal.

CHALLENGES AND OBSTACLES

The project group members are familiar with machine learning concepts and have some prior experience through coursework with natural language processing. NLP was found to be very different from ML in multiple ways. This project employed pretrained models within which it was challenging to affect a positive performance improvement. DistilBERT and other mainline NLP models are very capable out of the box and require a lot of experimentation to achieve better than baseline performance. Exploration of the model parameters did not reveal obvious combinations as the baseline had settled at the peak possible performance given baseline configuration.

The baseline tutorial Jupyter notebook did not appear compatible with the Kaggle runtimes out of the box. Some adjustments were needed to the tutorial Jupyter notebook to allow it to run; these were centered on manual selection of package versions (Keras-nlp, NumPy) changing an optimizer from TensorFlow versus Keras. These modifications allowed the baseline notebook to execute. The team eventually abandoned execution on Kaggle falling back to local Anaconda environments.

Also, there appears to be more variability in results for the same notebook run in similar environments. This could be due to slight differences in package versions or even as significant as differences due to CPU / GPU architecture. Differences in overall performance on Kaggle appear to be within 3/100th's of a percent. Note that random_state was set.

DISCUSSION AND FUTURE WORK

The team believes that NLP models are incredibly powerful affording understanding and application of large corpus to real world applications. Application of NLP models is different than other ML approaches requiring different techniques. Additionally, DistilBERT required greater CPU (and GPU) resources and runtime versus other ML models.

The team attempted several approaches to application of NLP through experimentation and learned a few things that worked and many that did not. This tracks with the level of effort invested by modern NLP entities such as OpenAI, DeepMind, Meta, and xAI to name a few.

The team is interested in future experimentation especially Retrieval Augmented Generation (RAG) and other modern and emerging approaches. These were briefly covered in prior coursework in COSC-524 but warrant further exploration.

SUMMARY

This project provided the opportunity to compare the performance of different machine learning and natural language processing models. A consumable data set and a challenge issued in the form of a competition provided an enriching and rewarding experience for the project group.

REFERENCES

[1] "Natural Language Processing With Disaster Tweets," [Online]. Available: https://www.kaggle.com/competitions/nlp-getting-started.

[2] "KerasNLP starter notebook Disaster Tweets," [Online]. Available: https://www.kaggle.com/code/alexia/kerasnlp-starter-

notebook-disaster-tweets.

[3] “NLTK improved stop words list” [Online]. Available https://gist.github.com/sebleier/554280.