EDU Categorizer

for

Transparent Text Classification

Ekrem Guzelyel, Hasan Rizvi, Anneke Shohrat Hidayat, Mustafa Bilgic

Illinois Institute of Technology

EDU Categorizer for Transparent Text Classification

Machine Learning applications have skyrocketed since the last 10 years. We now can predict the behavior and possible outcomes of a model. However, the reason behind those machine decisions is still an enigma. Researchers in Machine Learning has an increasing emphasis on this intriguing topic: transparency.

One research on role of human in interactive machine learning suggests that transparency can help people provide better labels (Amershi, et al., 2014). Kate Crawford in her NIPS 2017 keynote speech states that "the data that has been collected in a non-transparent way … can end up either in exclusion or over representation of a subpopulation" (Crawford, 2017). In our research we use EDUs as the back propagation method. EDU stands for the smallest unit of the sentence that makes a logical sense by itself.

Transparency is the end goal of the research in the big picture. However, before reaching this goal, in the small scale, we first have to have a solid amount of data, and an efficiently working model. Our task in this research is to find the best performing machine learning model that identifies EDUs in order to automatically generate more EDU labels.

Interactive Learning Application on Data Labeling

We need a big amount of data in order to train more complex models. Conversely, the cost of obtaining this data is proportional to the need, time wise. Therefore, applying interactive learning to the EDU pool and labeling only the EDUs that are likely to be harder to classify both gave more confidence to the models and decreased the amount effort to label the data.

Data

In the beginning we had 1980 data points, which are EDUs labeled as positive, negative or neutral. These are manually labeled by human labelers in the past. The EDUs are taken from reviews on IMDB movie dataset, and obtained after applying ABC technique of a Stanford professor (ABC, 2015).

In the end we have 5361 EDU labeled, of which around 1700 are negative, 1600 are positive, and 2000 are neutral. We only take negative and positive labels to make prediction, therefore we have 3300 data point to use in our model. Although it looks like a big amount, once we increase the complexity of a model, we see that it is not enough.

Approaches

I have tried 7 different models for prediction. I used Logistic Regression, Support Vector Machines, Multinomial Naïve Bayes, Long-Short Term Memory, Stacked LSTM, LSTM and CNN, and Gated Recurrent Units. I tested models along with labeling data simultaneously. Therefore, my early approaches mostly consist of models that don't require huge data. The main metrics for evaluation is test and train accuracy.

Logistic Regression: I used logistic regression to set a baseline for our models. With the early data, I had 0.8847 train accuracy, and 0.7636 test accuracy. The model gave a promising result, since it can be improved by increasing complexity.

Support Vector Machines: SVM models gave the best performance out of all the models. I got 0.9401 train accuracy, and 0.7955 test accuracy. Justification to this approach may be, because the relation of a word being positive and negative is directly related to how different it from being neutral, the SVM line performs better.

Multinomial Naïve Bayes: The basic approach of NBM didn't perform better than baseline LR model. However, I got 0.8392 accuracy on the train set, and 0.7530 on the test set. The explanation would be simply because the relation between EDUs and documents is more complex than a basic approach.

Long-Short Term Memory: I mainly worked on LSTMs to see if any combination of it can give a reasonable result. However, the best accuracy I got is 0.8880 on train data and 0.6438 on test data. In theory, I expected better result from this approach, since LSTM takes newer and older words that are encountered into consider, so that it has a better grasp on the entire review. The reason why it didn't demonstrate a good performance is possibly due to not supplying its demand for data size.

Stacked LSTM: "Stacking LSTM hidden layers makes the model deeper, more accurately earning the description as a deep learning technique" (Brownlee, 2017). Yet, the statement didn't hold for the small dataset. With hope of getting an improved prediction power, I tried stacked LSTM model, and got a train accuracy of 0.9040 and a test accuracy of 0.5514. Comparing the result from earlier models, it can be assumed that the model has overfit; however, in practice it didn't even reach the enough prediction power.

LSTM and Convolutional Neural Networks: Hoping that putting a convolutional layer on top of an LSTM structure would allow the model to catch the similarity in the pattern of the data, I placed

LSTM and CNN, one after another. Although, I got better score than both vanilla and stacked LSTM, the results didn't get close to the baseline. I got 0.9400 accuracy on train, which is almost as good as the SVM model, but only 0.6678 accuracy on test. My conclusion after this test is the model works, nevertheless, there is not enough data to compensate this complex model.

Gated Recurrent Unit: GRU is a type of RNN that handles small data size better. It uses gates to determine which of the past information needs to be passed along to the future and which to forget (Kostadinov, 2017). With this approach, I got 1.0000 train accuracy and 0.7220 test accuracy, which shows and overfit and an accuracy that is smaller than the baseline. Though, it is promising in a sense that a less complex model behaved better. This significantly suggests that the data size is not big enough.

	Train Accuracy	**Test Accuracy**
LR	0.8847	0.7636
SVM	0.9401	0.7955
MNB	0.8392	0.7530
LSTM	0.8880	0.6438
CNN+LSTM	0.9040	0.5514
Stacked LSTM	0.9400	0.6678
GRU	1.0000	0.7220

Table 1: Train and test accuracies of tested models.

Result

Trying all different approaches, the best performed model is a support vector machine approach. We get 0.9401 train accuracy, while having 0.7955 test accuracy. The consecutive trials with different complex models show that the data size is not enough to handle the complexity.

Future Work

The work already showed some significant results, however, in order to reach better performances, one has to utilize from more outside resources. Reading and analyzing more papers and experimenting recent unorthodox approaches may give more satisfactory results. Diversifying methods by using neutral cases is expected to pan out models that behave better. The ongoing research only has positive and negative labels taken into account. Also, to solve the biggest problem of data size, more data should be labeled manually. This way the complexity of models can be increased without worrying about the corresponding data. On top of that outside tools like TensorBoard and Vizier can be used to visualize and automatically pick the best model to save time and expedite the process.

Bibliography

Amershi, S., Cakmak, M., Knox, W. B., Kulesza, T. (2014). Power to the People: The Role of Humans in Interactive Machine Learning. *AI Magazine*, 35(4): Winter 2014, 105-120

Brownlee, J. (2017, August 18). Stacked Long Short-Term Memory Networks. Retrieved from https://machinelearningmastery.com/stacked-long-short-term-memory-networks/

Crawford, Kate. (2017). Keynote: The Trouble with Bias. Retrieved from https://www.youtube.com/ watch?v=fMym_BKWQzk.

Kostadinov, S. (2017, December 16). Understanding GRU Networks. Retrieved from https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be