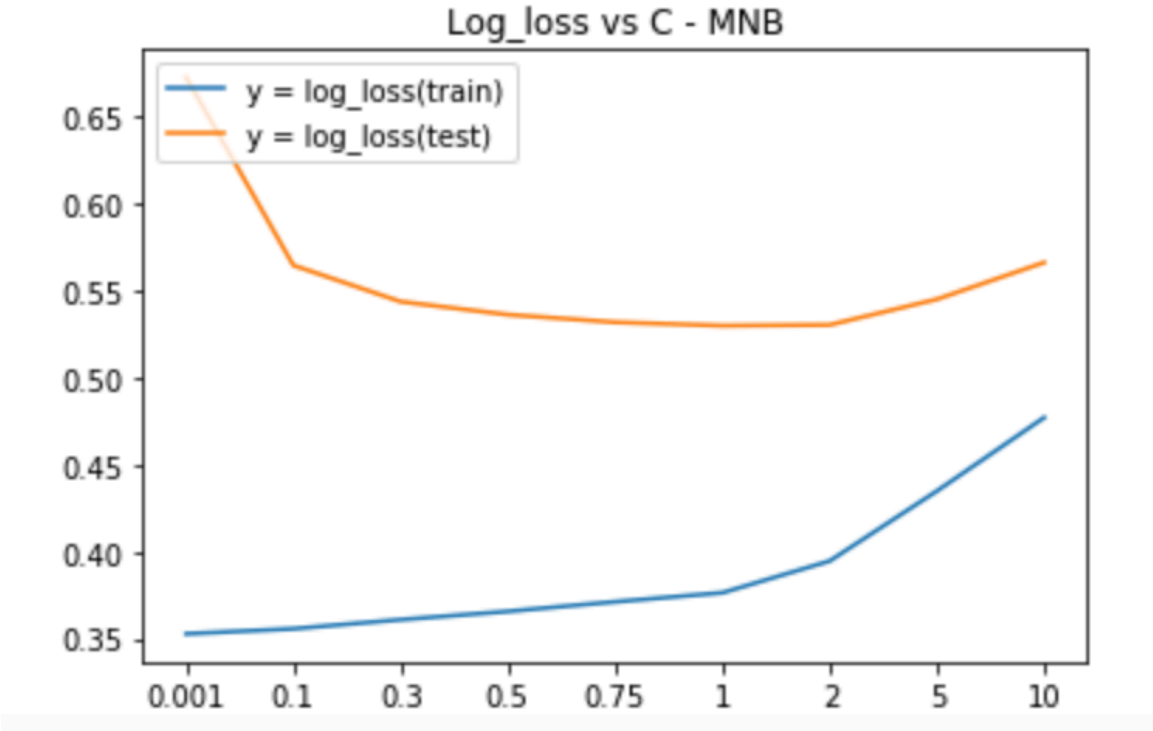
Machine Learning Laboratory at IIT - Fall 2018 Final Report

Transparent Text Classification

URA: Ekrem Guzelyel
Advisor: Anneke Suraya Hidayat, Mustafa Bilgic



Introduction & Purpose

Data means prediction power. However, sometimes you can't have enough data only with human sources. In order to classify movie reviews by IMDB, we use EDUs. Naturally, labeling all EDUs by hand consumes too much valuable time. Our job as Undergraduate Research Assistants at ML Lab at IIT is to build an efficient model that predicts the labels for the EDUs. This way we would be able to have a bigger dataset to train the actual text classification model. Tasks include:

- Labeling more EDUs on top of already labeled 2000 data points.
- Trying different combinations of different approaches to train a model that maximize the performance.

Approaches

I have used 7 different models for prediction; all of which resulted with different accuracies. These models are with using Logistic Regression, Support Vector Machines (SVM), Multinomial Naïve Bayes (MNB), Long-Short Term Memory (LSTM), Stacked LSTM, LSTM on top of Convolutional Neural Networks (CNN), and Gated Recurrent Units (GRU).

"Try every combination. One will behave better." – Anneke Soraya Hidayat

We expect one model to give a better accuracy than the baseline LR model.

	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

Metrics

The main metrics for this task is comparing train and test accuracy, along with looking at precision and recall. For some cases that were needed to be investigated deeper, log-loss has also been used.

"If you don't collect any metrics, you're flying blind. If you collect and focus on too many, they may be obstructing your field of view." – Scott M. Graffius

Data

We used EDUs that are extracted from IMDB movie review dataset. EDU can be explained as the smallest parts of the sentence that by itself make sense. These were in txt format. Hasan refactored the code and the dataset so that they are easy to use and add new inputs. At first we had around 2900 labeled EDUs, whereas only 1900 was either negative or positive. I personally labeled around 2000+ more EDUs. In the end we have 5200+ EDUs labeled, of which is mostly balanced towards positive, negative and neutral labels.

"Looks like we don't have enough data." – Ekrem Guzelyel

Communication

Both of the undergraduate students worked separately on their own models. We met weekly and discussed our progress. We collaborated with each other for labeling and code sharing over Github.

"Did you push your changes to Github?" – Syed Hasan Rizvi

"No." – Ekrem Guzelyel

Logistic Regression

We used Logistic Regression as the baseline. It is a fast and reliable way of making basic predictions. In the best case, defining C as 1.5, I got 0.7636 accuracy on test, and 0.8847 on train data. *See LogisticRegression.ipynb.*

"We use Logistic Regression as the baseline." – Every ML Researcher Ever

SVM

Support Vector Machines is also examined to find out if it gets a better accuracy. After looking at different kernels and gamma, I got the best result with using 'rbf' as the kernel, 2 as C value, and 0.1 for the gamma. As a result, the train accuracy was 0.9401, while the test accuracy was 0.7955. I also tried linear and sigmoid kernels, which can be viewed here. *See SVM.ipynb.*

"That's an interesting idea." – Caner Komurlu

MNB

I looked at Multinomial Naïve Bayes with varying alpha values. Along with accuracy metric, I checked precision and recall. However, it looked like the metrics didn't improve much with differing alpha value. On the best case, I used alpha as 1, and got test accuracy of 0.7530, and train accuracy of 0.8392. *See MNB.ipynb.*

"Stop, and take your time to breathe." – Mustafa Bilgic

LSTM

One of the problems that started to arise while looking at the LSTM models was the data size. I figured out that the existing number of labels weren't enough to make a reasonable prediction for a deep learning model. Using only 1 LSTM layer I got test and train accuracy of 0.6364 and 0.8292. After labeling some more data, I tried LSTM again, and this time I got a slightly better test and train accuracy of 0.6438 and 0.8880 with using 2 LSTM layers and 1 Dense layer. *See LSTM.ipynb.*

"Did you try ConvLSTM layer?" – Anneke Soraya Hidayat

CNN+LSTM

One other approach I used after discussing with professor is using an LSTM layer after a convolutional layer. I tried different combinations of CNN layers and with 1 LSTM layer on top. Oddly, the results weren't as good as expected. My theory is that we don't have enough data to support the complexity of the model. Although the idea was promising, the results stayed at 0.5514 test accuracy and 0.9040 train accuracy, which is the worst among all other models. *See LSTM_with_diff_layers.ipynb*

"Ask Jay, he knows about CNNs better." – Ruo Zhao

Stacked LSTM

Stacked LSTM models tend to behave better than one plain LSTM layer. I tested this theory, and proved it correct. I got the accuracy of 0.6678 on test and 0.9400 on train data. It's worth noting that I used embedding layer for all DL models. *See LSTM_with_diff_layers.ipynb*

"King minus man plus woman equals Queen" – Word2Vec Embedding

GRU

Gated Recurrent Units are the solution to lesser data size. After running with severel GRU units and Dense layers, I found out that the less complex the model, the better the accuracy. In the best case with only one GRU layer, I 0.7220 test accuracy and 1.0000 train accuracy. Though, train accuracy doesn't necessarily show that it overfit too much. The precision and recall scores show that the model has a reasonable prediction power. *See GRU.ipynb*

> *"Aim for simplicity in Data Science. Real creativity won't make things more complex. Instead, it will simplify them." – Damien Duffy Mingle*

Results

Overall, the model that performed the best was Support Vector Machines; the second was Logistic Regression, and MNB, GRU, Stacked LSTM, LSTM, LSTM+CNN respectively. We can infer that for this dataset the best model is SVM. However, this doesn't show that SVMs could be used as the ultimate model of the project. The results suggest that when a more complex model is used, it is more likely to give less accuracy. This points to only one result: We need more data. All of the models' results can be found here. Future work includes:

- More labeling is needed.
- Finding different approaches to training, so that the models fit to small datasets.

Summary: I have tried running the model using Logistic Regression, Multinomial Naive Bayes, SVM, LSTM with embedding layer, Stacked LSTM, and CNN+LSTM Although deep learning models are more complex, they didn't result better than simple models like LR,MNB and SVM. The reason is most likely the data size used. The labeled_edus.txt file has around 2000 negative/positive lines that can be used for classification. The best of each model and their parameters are shown below:

Model	Parameters			Test Accuracy	Train Accuracy	Notes
Logistic Regression	C=1.5	-	-	0.7636363636	0.8846737481	Recent runs result different than initial ones
MNB	alpha=1	-	-	0.753030303	0.8391502276	
SVM	C=2	kernel=rbf	gamma=0.1	0.7954545455	0.940060698	
LSTM (old edus)	#HL=1	LSTM_node=32		0.63636	0.8292	Embedding layer is used
LSTM (new edus)	#HL=2	LSTM_node=32	dense_node=20	0.6438356164	0.8880000001	Embedding layer is used
LSTM then CNN	Emb, LSTM(32)	CNN(128,5) CNN(128,5)	Batch=20	0.5509478672	0.9039145911	
Stacked LSTM	Emb	LSTM(32) LSTM(32)	Dense(10)	0.6678082189	0.9400000003	
CNN then LSTM	Emb CNN(128,5)	LSTM(32)	Batch=20	0.551369863	0.9039999998	In theory, this should've worked better
GRU	Emb	GRU(32)	Batch=20	0.7219647823	1	Less complex, better result

The best performance has gotten from SVM models.

What I Have Learned

I gained so much valuable information and experience from the project. To build each model, I had to read about the structure of the layers, and learn the idea behind it. In one semester, I learned and took a good grasp on the Machine Learning fundamentals and different methods. I learned the most about LSTMs and CNNs. I liked working in a group, I learned how communication is important.

"Communication!" – Anneke Soraya Hidayat

Special Thanks

I thank Ann to always help me when I am stuck, and answer all of my questions in a level I could understand. I thank Hasan to help me with coding tricks, and overall being an amazing lab partner. I thank Professor Bilgic for giving this opportunity. I thank Caner and Ruo for the laugh and energy they provide to the lab.

"Thank You!" – Ekrem Guzelyel