Toxic Post and Comment Detection

Chao Hsiang (Sean) Chung

Computer Sciences
University of Wisconsin - Madison
Madison, WI 53706, USA
seanchung@cs.wisc.edu

Abstract. There are boundless posts from Facebook, Twitter, Instagram, and so on. However, some of them might include violence, obscene, threat, insult, etc. In order to efficiently identify and remove these posts and/or messages which violate the terms of service, we need some trained models to support spotting them out.

In this paper I present a combination of models which use Long Short-Term Memory, Naive Bayes-Support Vector Machine, Convolutional Neural Network to identify if a comment is toxic.

1 Problem Statement

1.1 Related Works

Long Short-Term Memory (LSTM) Instead of doing sentiment analysis on specific keywords, it is better to analyze a sentence by its context. The motivation behind the LSTM model is that there can be lags of unknown duration between important events (contextual connections) in a time series (sentence). Also, a word appears in the beginning may somehow determine the meaning for another word later in a sentence or the entire meaning of a sentence. E.g., "am" decides the meaning of "I am a human", while "have" decides the meaning of "I have a human".

Naive Bayes-Support Vector Machine (NBSVM) For NBSVM, we want to first generate Term Frequency-Inverse Document Frequency (TF-IDF) where TF is the frequency of any word, and IDF is $\log(\frac{\text{number of documents}}{\text{number of documents which have the word in TF}})$. By doing TF-IDF, it can filter some frequent but meaningless words (e.g., I, this, etc.) since TF might be large, but IDF will be almost zero, which leads TF-IDF to nearly zero.

After doing TF-IDF, calculate conditional probability (parameters for Bayesian classifier) and derive the output from Bayesian classifier. These outputs will be data for support vector machine to compute the final prediction.

Convolutional Neural Network (CNN) For CNN, try to convert a sentence into a 2-Dimensional matrix to perform 2-Dimensional convolution. By performing 2-Dimensional convolutional and maxpool operations, it will generate the predicted result of the text/sentence.

1.2 Performance/Cost Metrics

The loss[1] and accuracy for these related works are:

Model	Training Accuracy	Training Loss	Testing Accuracy	Testing Loss	Kaggle Score
LSTM	0.9825	0.0475	0.9780	0.0736	0.9269
NBSVM	0.9983	0.0298	0.9485	0.0903	0.9615
CNN	0.9854	0.0372	0.9823	0.0534	0.9788

2 Model & Result & Analysis

2.1 Data

Data set is derived from "Toxic Comment Classification Challenge" on Kaggle. For each comment/post, it has six corresponding classes (labels):

- Toxic
- Severe Toxic
- Obscene
- Threat
- Insult
- Identity Hate

The objective for this task is to identify if a comment/post belongs to any of these classes (can be mutiple).

2.2 Methods

LSTM Before feeding data from raw dataset (sentences) into the model, it has to be preprocessed by

a. Tokenization, which breaks down the sentence into unique words. E.g., "AI is the future and ML is the future" becomes ["AI", "is", "the", "future", "and", "ML"]

[1] The loss function used here is binary cross entropy since we are predicting either it is toxic or not. Using binary cross entropy can be better than using mean square error

b. Indexing, which makes the words in a dictionary like structure and assigns each of them an index. E.g., {1: "AI", 2: "is", 3: "the", 4: "future", 5: "and", 6: "ML"}
c. Text to Sequence, which represents the sequence of words in the comments in the form of index, and feed this series of index into LSTM model. E.g., the original sentence becomes [1, 2, 3, 4, 5, 6, 2, 3, 4]

After converting articles from text to sequence, they are represented using word embedding to reduce the model size and high dimensionality. The output of the Embedding layer is a list of coordinates of words in the (word) vector space. Thus, the distance of these coordinates can be used to detect relevance and context.

Next, feed these coordinates into LSTM layer. As discussed previously, LSTM can be an ideal model dealing with data with time series. The model I referred to uses one-directional LSTM (Fig. 1.), which means that former words can influence latter words, while latter ones do not affect former ones. As natural languages are usually inter-weaved, the improvement here can be made is to change one-directional LSTM to bi-directional LSTM (Fig. 2.)

Finally, collects outputs from LSTM layer and feeds them to a neural network. After a series of trials, when the model has 3-layers (each of them has its dropout layer), and uses rectified linear unit (ReLU) as activation function, it can achieve the highest score.

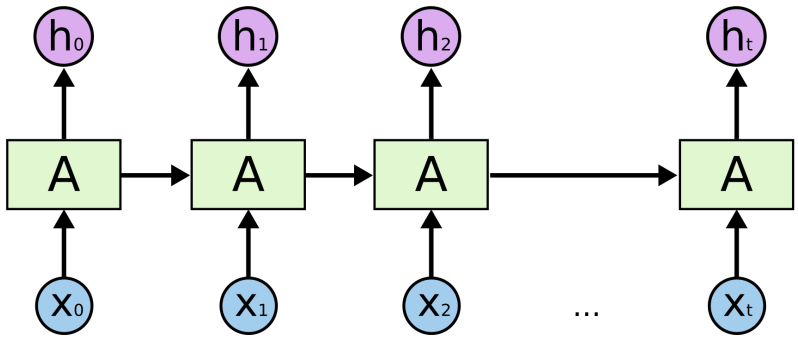

Fig. 1. As discussed, h_t will be affected by all X_{t^*}, where $t^* < t$ in one-directional LSTM. Namely, an output of a word is only influenced by past words.

NBSVM First, compute TF-IDF for each n-gram[2]. Using n-gram can link meanings of words in a sentence to better interpret a sentence.

[2] n-gram, also known as shingles, is a contiguous sequence of n words from a given sentence

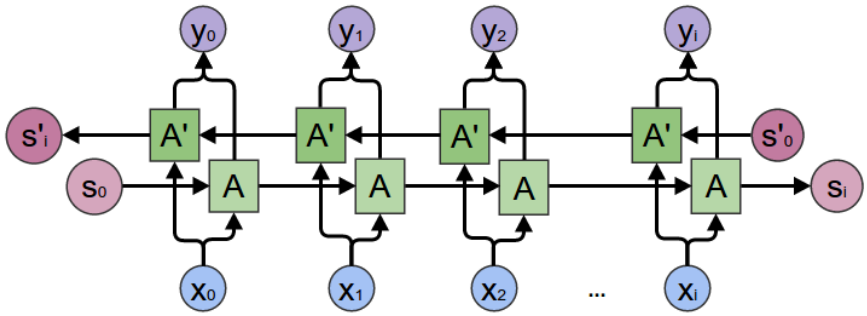

Fig. 2. Outputs of bi-directional LSTM y_t are influenced by x_{t^*}, where $t^* \neq t$. That is, for an output of a word, it is affected by past and future words.

After getting TF-IDF, calculate conditional probability for each word as parameters in a Bayesian classifier model, whose output will be data for training support vector machine (SVM) model.

Finally, train SVM model and outputs of the trained model are the prediction of whether the input sentence is toxic or not.

CNN Similar to LSTM, data has to be preprocessed by tokenization, indexing, text to sequence first. Next, data also has to be processed by embedding as LSTM.

The following is the procedure for doing CNN:

1. Processed data (sentence/paragraph) has to be converted to a 2-dimensional array (Fig. 3. (1)).
2. Apply several kernels on the 2-dimensional array. Kernel size should be n by the length of columns since n words should be examined together (similar to n-gram). (Fig. 3. (2))
3. Implement max pooling on each result from each kernel. The output of this layer will be one by one by depth. (Fig. 3. (3))
4. Concatenate and flatten outputs from previous layer, then feed the result into a one layer perceptron model. The final output is the prediction of whether it is a toxic comment. (Fig. 3. (4))

Ensemble Each of the models discussed above may be more or less biased on some specific types of data. In order to reduce variance and bias, and improve prediction, combining several machine learning techniques into one predictive model to balance different models. Here are two ways to implements ensemble:

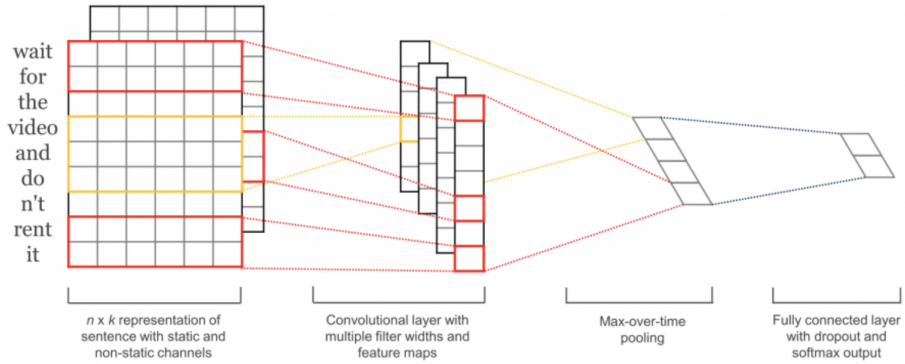

Fig. 3. (1) Convert a sentence/paragraph into a 2-dimensional array where each row represents a coordinate (vector) of a word. (2) Apply several kernels with size n by the length of columns. This is similar to n-gram. (3) Implement 2-dimensional max pooling on the outputs of step 2. and concatenate the outputs from max pooling layers to an one by m by depth tensor, where m is the number of kernels. (4) Flatten the tensor to a vector and feed it into an one-layer neural network to get the final result.

1. Average the predicted outputs.
2. Build another neural network to determine the final prediction by feeding in predictions from each model.

2.3 Model Structures

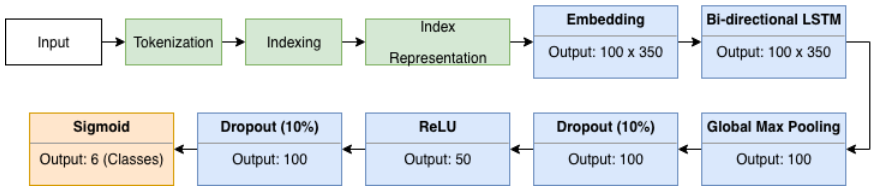

Fig. 4. LSTM Structure

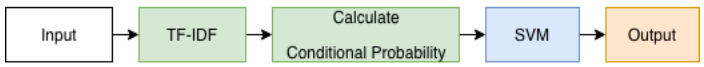

Fig. 5. NBSVM Structure

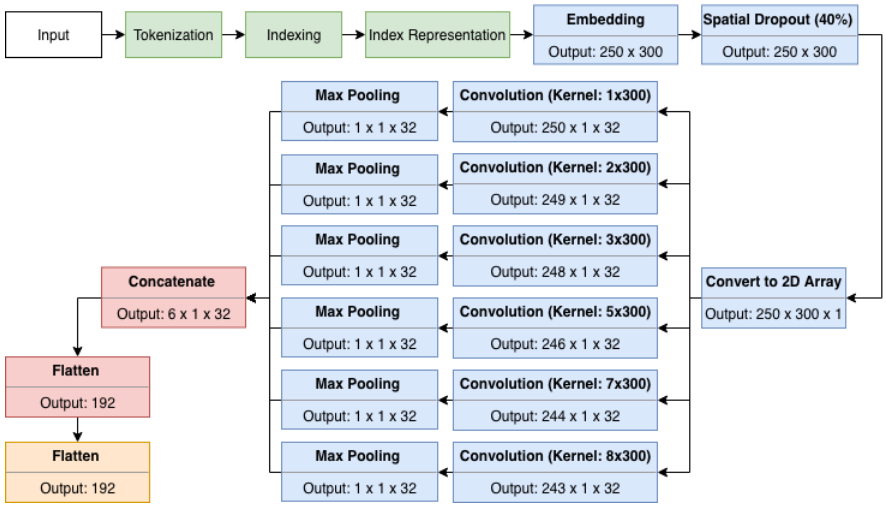

Fig. 6. CNN Structure

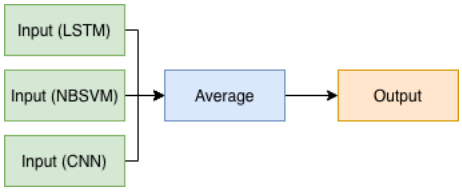

Fig. 7. Ensemble Structure

2.4 Results

Model	Training Accuracy	Training Loss	Testing Accuracy	Testing Loss	Kaggle Score
LSTM	0.9832	0.0449	0.9826	0.0492	0.9781
NBSVM	0.9994	0.0127	0.9547	0.0889	0.9645
CNN	0.9946	0.0154	0.9836	0.0427	0.9823
Ensemble	-	-	-	-	0.9834

2.5 Analysis & Discussion

After a few experiments, here are some values of parameters for LSTM, NBSVM, CNN, Ensemble models:

LSTM

- **Max features**[3]: 20000. For this data-set, when max features is set higher than 20000, the corresponding vector may be sparse; while when it is set lower, the corresponding vector will be too close. Both of them harms the results since they reduce the accuracy of measuring similarity between any two vectors.
- **Embedded Size**[4]: 350. Theoretically, the larger the size is, the more precise it maybe. I experimented from 100 to 600, but when it is around 350, the result is converged.
- **Dropout Percentage**: 10%. I experimented from 0% to 35%. When it is set to be 10%, it has the best result. It is due to the reason that when it is 0%, it is prone to over-fitting, while when it is higher than 10%, it will increase the bias, which causes lower performance.

NBSVM

- **N-gram**: 1-3. I experimented it with N=1-2, 1-3, 3-4, 3-5, 1-5, etc.. Unfortunately, due to my laptop capacity, it is not able to run N=1-5 or above. Among them when n=1-3 has the highest performance. The reason behind this may owing to the fact that the shorter length of a sub-sentence accounts for more importance than longer ones.
- **C (Regularization)**: 4. As we know, small C allows constraints to be easily ignored, which causes large margin; large C makes constraints hard to ignore, which causes narrow margin; when $C \to \infty$ enforces all constraints, which leads to hard margin. After a few experiments, when C is small (4), it has the best result since if C is too large, the SVM cannot find the separator; if C is too small, it will allow too many errors.

CNN

- **Max features**: 100000. Since there is an existing trained vector space, we can convert words to a larger space compared to LSTM's max features.
- **Embedded Size**: 300. Same as LSTM. I experimented from 100 to 600, but when it is around 300, the result is converged.
- **Kernel Sizes**: (1, 2, 3, 4, 5, 6, 7, 8) $\times$ (Embedded Size). As Fig. 3. (1), the width of the kernel has to be the same as embedded size, and the length of the kernel will be similar to n-gram. The reason CNN can have longer size than NBSVM is that it convolutes with larger size and it does not need to extract and replicate data again and again from the comment/post.

[3] The maximum number of different words in word-space
[4] The size of the vector for each word in word space

Ensemble

- **Neural Network** I implemented one-layer and multi-layer perceptron to emsemble results from LSTM, NBSVM, CNN. However, their performances are even lower than every result individually. The result may be caused by overfitting. After a series of trials, I chose not to continue with using this method due to low performance.
- **Average** When the results from LSTM, NBSVM, CNN are averaged, the performance measured by Kaggle score increases. I also experimented them with different weights, but mostly when they have equal weights, they perform better. Thus, my final ensemble model is to equally average the input.

3 Conclusion & Future Work

From this project, I learned how to apply neural networks to process natural languages. Measured by Kaggle score, from my original adopted model, with score of only 0.9269 (Rank: 4000+), to my improved model, with score of 0.9834 (Rank: 1702), it has advanced a lot. Considering the highest score is 0.9885, my model is quite close to the best one; however, I believe there is still little room for improvement. Here are some proposed solutions:

1. Ensemble with some more models (e.g., Gated Recurrent Unit, Hierarchical Attention Network, etc.) to mitigate bias.
2. Modify model structures (e.g., add mutiple LSTM layers to LSTM model)

Finally, apart from testing data, using my proposed approaches, my model can accurately identify real toxic comments/posts. Therefore, the model is useful.

References

1. Kiperwasser, E., & Goldberg, Y. (2016). Simple and Accurate Dependency Parsing Using Bidirectional LSTM Feature Representations. TACL, 4, 313-327.
2. Sundermeyer, M., Schlter, R., & Ney, H. (2012). LSTM Neural Networks for Language Modeling. INTERSPEECH.
3. Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. EMNLP.
4. Tackling Toxic Using Keras, https://www.kaggle.com/sbongo/for-beginners-tackling-toxic-using-keras.
5. NB-SVM strong linear baseline, https://www.kaggle.com/jhoward/nb-svm-strong-linear-baseline.
6. NLP using CNN, https://www.kaggle.com/rohitag13/bidirectional-gru-cnn