Jigsaw Unintended Bias in Toxicity Classification

Group members :
Ishak Shaik
Sanapala Sowmya
Krupa Kiranmai
Anisha Atyam
Mentor : Divya Acharya

Abstract

Social media is a platform which provide an environment where people can freely engage themself in discussions. Unfortunately, it leads to several problems too sometimes, such as online harassment. Recently, Google and Jigsaw started a project which uses machine learning to automatically detect toxic language. Machine learning models ate applied to identify toxicity in online conversations, where toxicity is defined as anything rude, disrespectful or otherwise likely to make someone leave a discussion.

Introduction

➢ Toxic comments contains foul language, derogatory remarks this could lead to:
- Spread of hatred
- Spread of racial slur
- Tension in communities
- Attack on individual
- It is abuse of freedom of speech

➢ Thus, this has to be monitored and censored on leading social networking sites .
➢ Jigsaw (subsidiary of Google) working on this problem developed 'Conversation AI'.
➢ To detect toxicity of a comment, it uses State-of-art methods like Deep Learning technologies.
➢Assigns score for toxicity to every comment.

Proposed Method

- ❑ Split the data to train, test, and validation.
- ❑ Train a word embedding model (word2vec/GloVe) on a dataset/pretrained model using gensim language modeling package
- ❑ Convert text to word-vector using the word embedding model.
- ❑ Build Classification Models on Train using:
- ✓ Recurrent Neural Nets – LSTM
- ❑ Evaluate on Validation data and tune the models. Repeat to find best parameters.
- ❑ Test performance of final model on Test dataset.

Experimental Results and Discussion

- Data - Preparation
  - Tokenize text
  - Convert words to index
  - pad the sequence - fixed size
  - replace index with pre-trained word vectors
- Final predictors (2 Dimensional array) - train Neural Nets
- crawl
  - Pre-trained model from Twitter
  - Contains 300-dimensional vectors for 3 million words
- gloVe
  - Pre-trained model from Stanford
  - Contains 300-dimensional vectors for 400,000 words
- keras word-embedding - Recurrent Neural Nets
- Early stopping using validation set.
- Tuning using:
  - Bidirectional-LSTM.
  - optimiser - Adam.

We used two methods for training a model with gradually increasing complexity of model.
In first phase we decided to use simple classifying algorithms which are computationally light and can serve as milestone for more complex methods.

Using logistic regression:

RESULT:

toxic comments

```
Test accuracy is 0.9216666666666666
```

severe_toxicity comments

```
Test accuracy is 0.9883333333333333
```

obscene comments

```
Test accuracy is 0.9516666666666667
```

threat comments

```
Test accuracy is 0.9966666666666667
```

insult comments

```
Test accuracy is 0.9516666666666667
```

identity_attack comments

```
Test accuracy is 0.9933333333333333
```

Using LSTM:

RESULT:

```
Epoch 1/1
 - 778s - loss: 0.5262 - dense_7_loss: 0.4196 - dense_8_loss: 0.1066 - dense_7_acc: 0.6937
- dense_8_acc: 0.8546
Epoch 1/1
 - 778s - loss: 0.5076 - dense_7_loss: 0.4055 - dense_8_loss: 0.1020 - dense_7_acc: 0.6954
- dense_8_acc: 0.8549
Epoch 1/1
 - 776s - loss: 0.5018 - dense_7_loss: 0.4007 - dense_8_loss: 0.1011 - dense_7_acc: 0.6957
- dense_8_acc: 0.8550
Epoch 1/1
 - 774s - loss: 0.4981 - dense_7_loss: 0.3975 - dense_8_loss: 0.1006 - dense_7_acc: 0.6959
- dense_8_acc: 0.8550

Complete. Exited with code 0.
```

Conclusions

- In this project we successfully employed word2vec embedding and recurrent neural network in building a toxic comment classification model and achieved high accuracy with relatively low cost.
- We can also perform additional hyperparameter tuning on our model, which will most definitely prove beneficial.
- On a closing note, Conversation AI team's intention and effort in building an open source tool to monitor and control online toxicity is commendable.
- Researchers and discussion platform moderators have already found numerous ways to apply this tool in very creative manners.
- We hope that with the collaborative help from the machine learning community,the team can continuously improve the performance of Perspective API and help maintain a toxic-free environment for our online discussions.

References

Data References:
https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification/da
Github Code:
https://github.com/ishaks671/unintended-bias-in-toxicity-classification
YouTube Video:
https://www.youtube.com/watch?v=tLjgKcnjMW0&feature=youtu.be

Team No. - 40