“Its OK to be Black”

But is that okay to say?

Johan Ahlqvist
johan.ahlkvist@gmail.com

André Skoog
nyg15ask@student.lu.se

Unintended Bias in Toxicity Classification

- A competition by Google’s subsidiary Jigsaw
- Hosted on data science competition site Kaggle (check it out!)

The task

Given an out-of-context forum comment, classify whether it is toxic or not

The performance metric

- Previous toxic comment classification competition
    - The metric: AUC over toxicity types
        - Problematic: bias encouraged

“I am a gay woman” ⇒ ☑️ “gay” present in comment ⇒ toxic!

The performance metric cont'd

- New toxic comment classification competition
  - Google: "penalize bias!"
  - The resulting new metric: .. complicated
  - "Overall AUC plus generalized mean of bias AUCs"

$$M_p(m_s) = \left(\frac{1}{N}\sum_{s=1}^{N} m_s^p\right)^{\frac{1}{p}}$$

$$score = w_0 AUC_{overall} + \sum_{a=1}^{A} w_a M_p(m_{s,a})$$

M_p = the pth power-mean function

m_s = the bias metric m calulated for subgroup s

N = number of identity subgroups

A = number of submetrics (3)

$m_{s,a}$ = bias metric for identity subgroup s using submetric a

w_a = a weighting for the relative importance of each submetric;

The performance metric cont’d

- Basic idea: penalize poor classification performance on comments that contain identities
- Four components:
    - AUC of each subgroup on comments containing identities and identity toxicity
    - AUC of each subgroup on comments containing identities but not identity toxicity
    - AUC of each subgroup on comments where identities are present
    - Overall AUC on all comments

The experience

- 1.8 million comments from Civil Comments
- Metadata:
  - Is comment toxic?
  - What type of toxicity? (threat, insult, etc)
  - What identities are mentioned?

How do we feed a computer text?

~~How do we feed a computer text?~~

How do we train a neural network on text data?

Processing text for input to a neural network

- **Preprocessing**
- Tokenization
- Word embeddings

Preprocessing

Increase our chances of recognizing words

- Fix misspelled words (yuor -> your)
- Rewrite contractions (omg -> oh my god)
- Remove special characters (punctuation, smileys, etc)
- Set text to lowercase
- Separate punctuation

Not always a good idea, some information is inevitably lost

Processing text for input to a neural network

- Preprocessing: alter data to make it more easily recognizable
- **Tokenization**
- Word embeddings

Tokenization

- Transforming words to IDs according to a map
    - Normally mapping to an integer according to frequency
- Tokenization may be done at different levels, e.g.:
    - Sentence
    - Word
    - Character

Example of word tokenization:

the be to of and a in that have I it for not

0 1 2 3 4 5 6 7 8 9 10 11 12

Processing text for input to a neural network

- Preprocessing: alter data to make it more easily recognizable
- Tokenization: convert words to IDs
- **Word embeddings**

Word embeddings

- Translates IDs to feature vectors
- Feature vectors contains many numbers for each word that together describe the word's characteristics
    - Commonly, 50-300 dimensions per word are used
- Words with similar semantic meaning should have similar feature vectors
    - Vectors of dog and wolf more similar than dog and human

May look like:
Human 0.2483 0.6843 -0.6322 0.1828 -0.5912 ….

Processing text for input to a neural network

- Preprocessing: alter text to make it more easily recognizable
- Tokenization: convert words to IDs
- Word embeddings: convert IDs to feature vectors

Our solution

- **Regularization**
- Network architecture
- Inputs and outputs
- Ensembling

Regularization

- Using:
  - Spatial dropout (replace 20% of embedding feature maps with noise)
- Tested but rejected:
  - Weight decay
  - Batch normalization
  - Dropout

Our solution

- Regularization
- **Network architecture**
- Inputs and outputs
- Ensembling

Network architecture

- 300 dimensional GloVe Common Crawl embedding with 840 billion tokens
- Bidirectional LSTM
- Attention
- Fully connected layers with skip connections

Our solution

- Regularization
- Network architecture
- **Inputs and outputs**
- Ensembling

Inputs and outputs

- Inputs
    - The comment data
    - Statistics about what amount of caps/punctuation was used in the comment
- Outputs
    - Whether the comment is toxic
    - What type of toxicity is present in the comment
    - Whether the comment contains toxic use of identities

Our solution

- Regularization
- Network architecture
- Inputs and outputs
- **Ensembling**

Ensembling

- Two models were trained for 4 epochs each
- A prediction was made for each epoch
- Final prediction = weighted average over all predictions with the higher epochs given a higher weight

Results

- Score: 0.93597
- Current position on leaderboard: 542/2136

“Its OK to be Black”

- An actual sample that was misclassified as toxic by our model
- Likely caused by bias for the word “Black”

Moving forward

- Adding sentence context to word embedding
  - Words’ meaning depend on context
    - “Give me that stick”
    - “Stick to the plan”
      - Different sticks!
  - Possible by using BERT
- Making smaller models to enable larger ensembling within time limit

Thank you for your **attention**