Bypassing Censorship: Reverse Engineering a Social Media Censorship Classifier to Generate Adversarial Posts

Chris Cross, Sasank Munukutla, Tan Siah Yong
{chrisglc, sasankh, siahyyong}@stanford.edu

CS229: Machine Learning

Motivation
- China’s microblogging site, Weibo, often censors posts containing sensitive material
- Can we find a way to ignore its censorship filters?
- The classifier is then used to help generate posts that bypass censorship with word substitution
- Results show our approach mimics Weibo’s censor

China's microblogging site, Weibo, often censors posts containing sensitive material. Can we find a way to ignore its censorship filters? The classifier is then used to help generate posts that bypass censorship with word substitution. Results show our approach mimics Weibo’s censor.

Dataset / Preprocessing
- Dataset: 60,000 Weibo posts from Weiboscope1, labelled 1 or 0 for censored
- Preprocessing: Removed links, numbers, special text, and post w/ < 5 characters
- Tokenization: Used SnowNLP2 to segment words and remove stopwords

Dataset / Preprocessing
- Dataset: 60,000 Weibo posts from Weiboscope, labelled 1 or 0 for censored
- Preprocessing: Removed links, numbers, special text, and post w/ < 5 characters
- Tokenization: Used SnowNLP to segment words and remove stopwords

Traditional Models
- Multinomial Naive Bayes: with Laplace smoothing: multivariate bernoulli model for both BoW and TF-IDF
- Kernel SVM: Support vector machine with RBF kernel tuned for gamma and learning rates with K cross-fold validation
- Logistic Regression: Use a weighted, regularized logistic regression classifier of the form \( h(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}} \)

Traditional Models
- Multinomial Naive Bayes: with Laplace smoothing: multivariate bernoulli model for both BoW and TF-IDF
- Kernel SVM: Support vector machine with RBF kernel tuned for gamma and learning rates with K cross-fold validation
- Logistic Regression: Use a weighted, regularized logistic regression classifier of the form \( h(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}} \)

More Complex Models
- CNN: We used an embedding layer of Keras to learn word embeddings, and a convolutional layer w/ max-pooling
- LSTM-CNN: Building on the CNN we add an LSTM layer. Hyperparameters were tuned using GridSearchCV
- Light-GMB: Used similar features as traditional models, GridSearchCV to tune for learning rate, # leaves

More Complex Models
- CNN: We used an embedding layer of Keras to learn word embeddings, and a convolutional layer w/ max-pooling
- LSTM-CNN: Building on the CNN we add an LSTM layer. Hyperparameters were tuned using GridSearchCV
- Light-GMB: Used similar features as traditional models, GridSearchCV to tune for learning rate, # leaves

Model Training Results

Model Training Results

Features
- Bag-of-Words: Unigram word tokens, vectorized according to the frequency of each word in every post
- TF-IDF: normalizes the word frequencies in a post by how often each word appears in other posts
- Word2Vec: pretrained word embeddings with gensim module; vocab size: 50013, embed size: 300
- Sequence: default word embedding layer used by Keras for both the CNN and the LSTM-CNN

Features
- Bag-of-Words: Unigram word tokens, vectorized according to the frequency of each word in every post
- TF-IDF: normalizes the word frequencies in a post by how often each word appears in other posts
- Word2Vec: pretrained word embeddings with gensim module; vocab size: 50013, embed size: 300
- Sequence: default word embedding layer used by Keras for both the CNN and the LSTM-CNN

Future Work
- Greatly expand on experimental setup (far more examples split across more accounts during different time intervals)
- Improve either our classifier (GloVe embeddings, BERT, etc) or substitution algorithm (beam search, etc)
- Factor poster account (post history, followers) as features

Future Work
- Greatly expand on experimental setup (far more examples split across more accounts during different time intervals)
- Improve either our classifier (GloVe embeddings, BERT, etc) or substitution algorithm (beam search, etc)
- Factor poster account (post history, followers) as features

Homophonic Substitution Algorithm
- Replace words with randomly generated homophones
- Zhèng fù (government) \(\rightarrow\) Zhèng fū (no meaning)

Homophonic Substitution Algorithm
- Replace words with randomly generated homophones
- Zhèng fù (government) \(\rightarrow\) Zhèng fū (no meaning)

Experimental Setup and Preliminary Results

Experimental Setup and Preliminary Results

References
1. [https://weiboscope.jmsc.hku.hk/wsr/](https://weiboscope.jmsc.hku.hk/wsr/)
2. [https://github.com/microsoft/LightGBM](https://github.com/microsoft/LightGBM)

References
1. [https://weiboscope.jmsc.hku.hk/wsr/](https://weiboscope.jmsc.hku.hk/wsr/)
2. [https://github.com/microsoft/LightGBM](https://github.com/microsoft/LightGBM)