Sweet or Sour: Insult Detection Milestone

Anonymous Author(s)
Affiliation
Address
email

Abstract

Abusive comments are increasingly drowning out constructive ones, and websites are reacting by shutting down comment sections. Human moderation is slow and expensive, so an algorithmic solution would be preferable. In this project I explore the problem, existing literature, and several models to automatically flag abusive comments.

1 Intro

Online forums seem to be increasingly toxic. Last year alone, Bloomberg, The Verge, The Daily Beast, and Vices Motherboard all shut down their comment sections due to concerns about the tone and quality of comments. Abusive comments, then, can be considered an existential threat to online discourse.

Current solutions usually employ human moderators. The moderators then either must approve each comment before it appears, or review a comment when it has been flagged by a user. Both of these techniques have significant drawbacks: approving every comment is slow and discourages discussion, but waiting until a comment has been flagged means that, in some sense, the damage has been done. Both of these techniques are also finanially expensive as they require a real human to do the job.

An ideal solution would be a completely autonomous system to prevent abusive comments from ever being posted, but due to the imperfections of current systems it is more likely that flagged messages will be sent to human moderators. instructions below, and follow them faithfully.

1.1 Problem Decription

This project looks at insult detection, which is detecting when a comment is directly insulting a person in the conversation. This is different from true abusive comment detection in that it is narrower. While true abusive comments can be hate speech, harassing the author of an article, or insulting a participant - insult detection is only the final part.

The reason that I am restricting my focus is that there are currently no good abusive commment datasets. Yahoo will release one later in May [1], and I will try to run models when that comes out for comparison.

Nevertheless, insult detection is still an interesting problem because it requires real-world knowledge of what is insult and what is disagreement, and needs to differentiate censuring abuse from abuse.

1.2 Data

Until the Yahoo data is available [1], I will be using the Kaggle "Detecting Insults in Social Commentary" dataset. It contains a train set (3445 comments, 29% insults), validation (1494 comments,

Table 1: Preliminary Results

Model	F1
Baseline	0.73
GRU 100dim Bidirectional	0.60
GRU 200dim	0.66

27% insults), and test (1938 comments, 47% insults). I have merged the train and validation data, and split the test data to make my new (train/dev/test)

I manually inspected 50 comments from the original train and test data (100 total), and found a 4% error rate in the train and 14% error rate in the test. At the very least, this would be considered human-level performance and can be considered an upper bound on performance.

On the other hand, this is concerning because it suggests the data is not correctly labeled, and problematic in other ways. It is also concerning because it indicates the test set is not representative of the training set because they have very different error rates. I believe this is because the test set has more insults, and these labels are more subjective.

In either case, other papers have published using this data [2], and until the Yahoo data set comes out, this is the best option available.

2 General Approach

I will try to improve on this result by trying a character-RNN (or GRU, LSTM) with possible more than one hidden layer. Comments contain many misspellings, dropped spaces, and punctuation. I believe that a character RNN will be able to use this information to inform its classification. That being said, the best performance in the 2012 Kaggle competition was from linear models.

If time permits, I will also experiment with using word-level RNNs to classify. I believe that while the character-level approach has more theoretical merit, the amount of research done on distributional word approaches and superior initialization of word vectors will overcome the theoretical benefit. This is more likely as the amount of data is small.

2.1 Restricting the data

Because some comments are long (20000 chars) while 80% are under 200 characters, I will restrict all models to comments which are under 200 chars. This makes them easier to analyze without too great of a loss in power. I have not looked at the exact difference between long/short comments, but the baseline model had similar performance both with/without long comments.

3 Preliminary Results

I have set up a baseline using unigram+bigram logistic regression with some preprocessing of the text. This model places around 10/50 in the original 2012 Kaggle competition when using the original data set splits. The competition used AUC as the metric.

All the RNN models below use xavier initialization everywhere except the biases which are zero. They use ADAM lr=0.001. For regularization, they use dropout 0.9 after the output layer, used early stopping and insult upsampling.

1

I also tried two layers GRU, a 200-dim RNN, and a 100-dim bidirectional LSTM, none of them did significantly better. Unfortunately, I've been changing my checkpoint format and lost these results. I'll have to re-run the models.

It appears the model is over-fitting as GRU-200 scores 0.88 on the training dataset.

4 Next Steps

I will try to find the optimal hyperparameters for a forward LSTM/GRU. I will do a greedy search among with learning rate, dropout (and apply it other places, and to characters). Once I have these, I will experiment with other char-RNN models.

If time permits, I will also try word-rnn models which might be better. I think they might be because I have relatively little data, and can transfer the information from word vectors.

I also believe they will do better because the (word-level) baseline outperforms the RNNs now.

References

[1] Chikashi Nobata, Joel R. Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. Abusive language detection in online user content. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 145–153, 2016.

[2] Priya Goyal and Gaganpreet Singh Kalra. Peer-to-peer insult detection in online communities. *IITK, unpublished*, 2013.

[3] April Kontostathis, Kelly Reynolds, Andy Garron, and Lynne Edwards. Detecting cyberbullying: Query terms and techniques. In *Proceedings of the 5th Annual ACM Web Science Conference*, WebSci '13, pages 195–204, New York, NY, USA, 2013. ACM.

[4] Tatsuya Ishisaka and Kazuhide Yamamoto. Detecting nasty comments from bbs posts. In *PACLIC*, pages 645–652, 2010.

[5] D Sculley. *Advances in online learning-based spam filtering*. ProQuest, 2008.

[6] William Warner and Julia Hirschberg. Detecting hate speech on the world wide web. In *Proceedings of the Second Workshop on Language in Social Media*, pages 19–26. Association for Computational Linguistics, 2012.

[7] Amir H. Razavi, Diana Inkpen, Sasha Uritsky, and Stan Matwin. *Advances in Artificial Intelligence: 23rd Canadian Conference on Artificial Intelligence, Canadian AI 2010, Ottawa, Canada, May 31 – June 2, 2010. Proceedings*, chapter Offensive Language Detection Using Multi-level Classification, pages 16–27. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.