Toxic Comments Kaggle Competition

The Competition Results - *public leaderboard* top 10 contestants all between .9876 and .9885 AUC

In the money · Gold · Silver · Bronze

#	∆pub	Team Name	Kernel	Team Membe...	Score	Entr...	Last
1	—	Toxic Crusaders			0.9885	171	16d
2	—	neongen & Computer s...			0.9882	129	16d
3	▲3	Adversarial Autoenco...			0.9880	451	15d
4	▲1	Leyantech			0.9878	164	15d
5	▲2	TPMPM			0.9878	299	15d
6	▼3	Mike			0.9878	182	16d
7	▲1	GL Team			0.9878	247	15d
8	▲3	Lake Unanimated			0.9877	59	15d
9	▲1	TetyanaYatsenko			0.9877	240	15d
10	▼6	DecisionGuys		+4	0.9876	397	15d

The Competition

General Description https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/overview

- It would be helpful to be able to automatically detect toxic comments in online discussions
- In this competition one predicts whether each comment is one of 6 kinds of toxic comments.

Data - csv files - comments from wikipedia

training data: 159571 labelled comments

id	comment_text	toxic	severe_ toxic	obscene	threat	insult	identity _hate
0001d958c54c6e35	You, sir, are my hero. Any chance you rem	0	0	0	0	0	0
00025465d4725e87	"	0	0	0	0	0	0
0002bcb3da6cb337	COCKSUCKER BEFORE YOU PISS AROUND O	1	1	1	0	1	0
00054a5e18b50dd4	bbq	0	0	0	0	0	0
0005c987bdfc9d4b	Hey... what is it.. @ | talk . What is it... an exclusive group of some WP TALIBANS...who are good at destroying, self-appointed purist who GANG UP any one who asks them	1	0	0	0	0	0
0006f16e4e9f292e	Before you start throwing accusations	0	0	0	0	0	0
00070ef96486d6f9	Oh, and the girl above started her argume	0	0	0	0	0	0
00078f8ce7eb276d	"	0	0	0	0	0	0
0007e25b2121310b	Bye! Don't look, come or think of comming back! Tosser.	1	0	0	0	0	0
000897889268bc93	REDIRECT Talk:Voydan Pop Georgiev- Cher	0	0	0	0	0	0
0009801bd85e5806	The Mitsurugi point made no sense - why	0	0	0	0	0	0

Test data: 153164 unlabelled comments

id	comment_text
00001cee341fdb12	Yo bitch Ja Rule is more succesful then you'll ever be whats up with yo
0000247867823ef7	== From RfC ==
00013b17ad220c46	"
00017563c3f7919a	:If you have a look back at the source, the information I updated was t
00017695ad8997eb	I don't anonymously edit articles at all.
0001ea8717f6de06	Thank you for understanding. I think very highly of you and would not
00024115d4cbde0f	Please do not add nonsense to Wikipedia. Such edits are considered v
000247e83dcc1211	:Dear god this site is horrible.
00025358d4737918	"
00026d1092fe71cc	== Double Redirects ==
0002eadc3b301559	I think its crap that the link to roggenbier is to this article. Somebody t
0002f87b16116a7f	"::: Somebody will invariably try to add Religion? Really?? You
0003806b11932181	, 25 February 2010 (UTC)
0003e1cccfd5a40a	"

Submission Format - A CSV file:

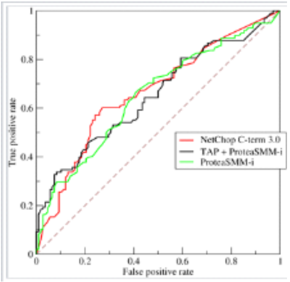
```
id,toxic,severe_toxic,obscene,threat,insult,identity_hate
00001cee341fdb12,0.9950,0.1934,0.9640,0.0301,0.8955,0.1339
0000247867823ef7,0.0013,0.5116,0.0002,0.0001,0.0005,0.0001
00013b17ad220c46,0.0047,0.0003,0.0006,0.0006,0.0017,0.0006
00017563c3f7919a,0.0020,0.5179,0.0003,0.0001,0.0005,0.0002
```

id	toxic	severe_ toxic	obscene	threat	insult	identity_ hate
00001cee3	0.995	0.1934	0.964	0.0301	0.8955	0.1339
00002478€	0.0013	0.5116	0.0002	0.0001	0.0005	0.0001
00013b17a	0.0047	0.0003	0.0006	0.0006	0.0017	0.0006
00017563c	0.002	0.5179	0.0003	0.0001	0.0005	0.0002

Evaluation Criteria - the mean column-wise ROC AUC

- The areas under the ROC curve for each type of toxic comment, averaged

An example ROC curve



Interpretation: In order to correctly identify more of the positives, we will also mis-label more of the negatives as positive. i.e.:

In order to identify 40% of positives, we will mis-label ~20% of negatives as positives

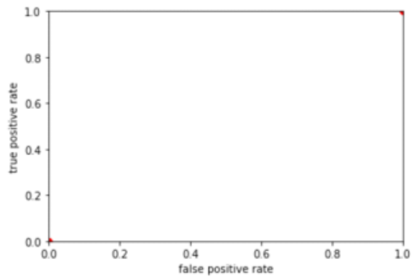
In order to identify 80% of positives, we will mis-label ~60% of negatives as positives

How do we get a ROC curve from a submission?

Example labels and predictions:

Id	Is toxic	Prediction (probability toxic)
1	False	.05
5	False	.1
4	False	.2
2	True	.15
3	True	.35

Let's make the ROC curve…



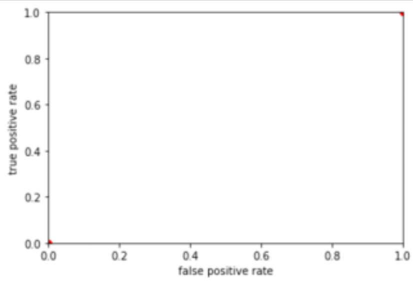
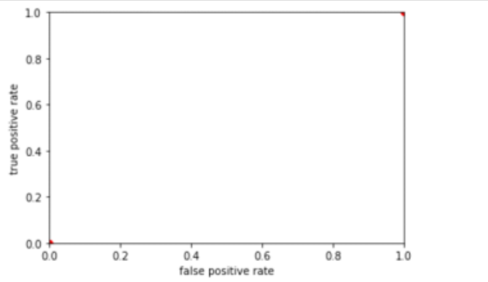
Calculating it's ROC curve

Id	Is toxic	Prediction (probability toxic)	Pred. if 0.0 threshold	Pred. if 0.07 threshold	Pred. if 0.12 threshold	Pred. if 0.17 threshold	Pred. if 0.3 threshold	Pred. if 0.5 threshold
1	False	.05	True	False	False	False	False	False
5	False	.1	True	True	False	False	False	False
4	False	.2	True	True	True	True	False	False
2	True	.15	True	True	True	False	False	False
3	True	.35	True	True	True	True	True	False

Optimizing the area under the ROC curve

- *This method optimizes the ROC AUC, given the probabilities you have available to you.*
- *Hence, **we could just submit the best probabilities we can.** They'll calculate the optimal ROC via this method for us.*
- *However, if our probabilities are all off (i.e. overconfident from overfitting), without changing their relative ordering, this won't affect the ROC curve or our predictions.*

What would the ROC curve look like if…:

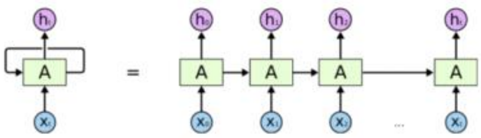
You had a perfect classifier?	You made up random probabilities of being invasive?
	

Theory Background - *Gru, BiGru*

[*link*]

RNNs (Recurrent Neural Network)

Great for sequential data



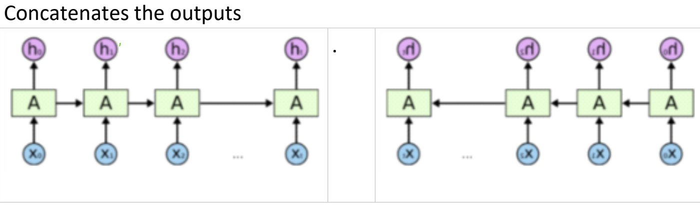
An unrolled recurrent neural network.

Bi-Directional RNNs

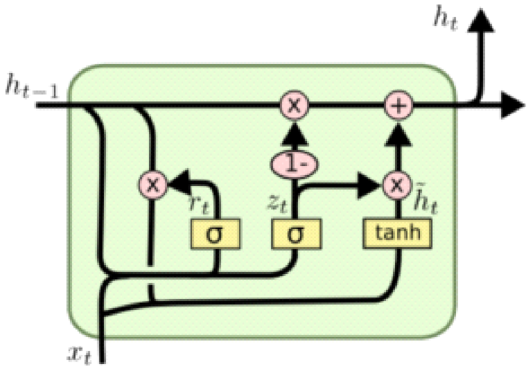
Runs one RNN forward over the sequence

Runs one RNN backward over the sequence

Concatenates the outputs

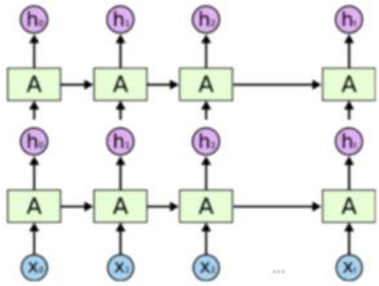


Gru, BiGru



The 'X' are the 'gates', which control how much is 'forgotten' then 'replaced'
This allows the GRU to retain state for long periods of time until it is needed.
The BiGru just runs one GRU in each direction over the sentence

Multiple layers of RNNs (Usually 2 in this competition)



The First Place approach

They have an excellent write-up

Basic model:

i. Word embeddings
ii. Two BiGru layers
iii. Two dense layers
iv. Output

a. Diverse pre-trained embeddings (baseline public LB of 0.9877)
   - >90% of a model's complexity resides in the embedding layer
   - Used the highest-dimensional FastText and Glove embeddings pre-trained against Common Crawl, Wikipedia, and Twitter
b. Translations as train/test-time augmentation (TTA) (boosted LB from 0.9877 to 0.9880)
   i. Used French, German, and Spanish translations translated back to English
   ii. Used for training, and test
c. Rough-bore pseudo-labelling (PL) (boosted LB from 0.9880 to 0.9885)
   i. Labelled the test data with best ensemble, then
   ii. Trained on that
d. Robust CV + stacking framework (boosted LB from 0.9885 to 0.9890)
   - Used a mix of arithmetic averaging and LightGBM

Other take-aways
- Since most of the model complexity lay in the pre-trained embeddings, minor architecture changes made very little impact on score.
- Our best CNN (a wavenet-like encoder connected to some time distributed dense and dense layers) scored about .0015 lower than our best RNN.

What approaches won - A comparison

Placing	Score	Architectures Enssembled	Approaches Used

1st	.9885	RNN (BiGru)	[x] Diverse pre-trained embeddings
			[x] train and test-time augmentation (TTA) using translations to other languages and back.
			[ ] Train on translation
			[x] Pseudolabelling
			[x] Enssembling
			    [x] Averaging
			    [x] Stacking
			[ ] Feature engineering
			[ ] Training own embeddings for OOV words
			-- 	

link

neongen
2nd place
	.9882	RNN, DPCNN and GBM	[x] Diverse pre-trained embeddings
			[x] train and test-time augmentation (TTA) using translations to other languages and back.
			[x] Train on translation
			[ ] Pseudolabelling
			[x] Enssembling
			    [x] Averaging
			    [ ] Stacking

			☐ Feature engineering ☐ Training own embeddings for OOV words --

link

Bojan Tunguz
3rd place
	.9880	GRU, LSTM and GRU + CNN
		
		tf-idf vectorizations with Logistic Regression
		
		XGBoost
			☑ Diverse pre-trained embeddings
			☑ train and test-time augmentation (TTA) using translations to other languages and back.
			☑ Train on translation
			☐ Pseudolabelling
			☑ Enssembling
			    ☑ Averaging
			    ☐ Stacking
			☑ Extensive Feature engineering (spelling, all caps words)
			☑ Training own embeddings for OOV words
			-- 	

General Take-Aways:

- Ways to add extra information (aside from the original features and labels):
  - Transfer learning is very important
    - □ Usually from pre-trained embeddings (larger generally better)
    - □ Can train own embeddings for OOV words
    - □ Pre-trained embeddings were not trained further to prevent fitting.
  - Feature engineering can make a big difference (adding human knowledge)
  - Effective data-augmentation for text: Translate to other languages then back to English
- Enssembling - including over the Pre-trained embeddings used!

•