Deep Learning Humor Classification on Yelp Reviews

CS 224N

Rohan Bais & Marcos Torres

Stanford
Computer Science

Problem

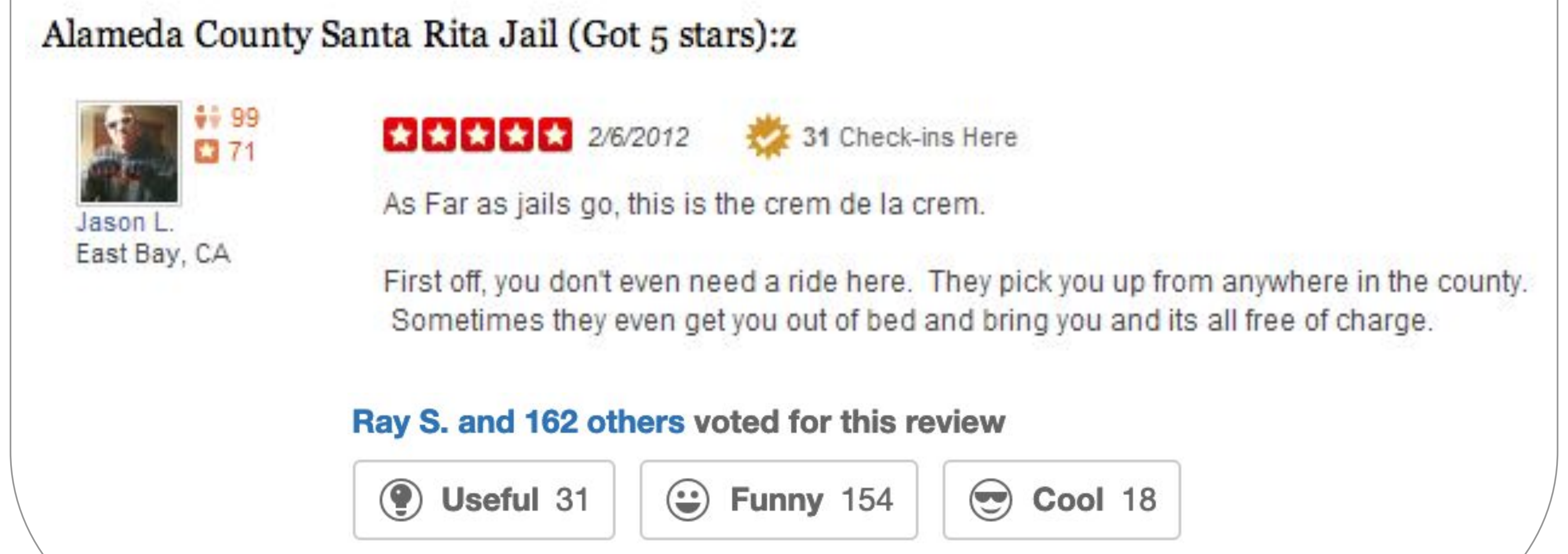
Humor is an abstract, high-level use of language that is largely subjective. Still, are structures and patterns of language that are widely recognized as funny.

We attempt to build a model capable of recognizing humor within text. Specifically, we're interested in humor as it arises unforced in everyday situations – that is, not scripted like a sitcom.

While a light-hearted endeavor, this work has potential applications in making personal assistants more responsive and life-like.

Dataset

We use 120,000 yelp reviews, each labeled with a count of "funny" votes. The text was tokenized and padded. We assigned a positive label if a review had at least one "funny" vote.



Alameda County Santa Rita Jail (Got 5 stars):z

Jason L.
East Bay, CA
99
71

★★★★★ 2/6/2012 31 Check-ins Here

As Far as jails go, this is the crem de la crem.

First off, you don't even need a ride here. They pick you up from anywhere in the county. Sometimes they even get you out of bed and bring you and its all free of charge.

Ray S. and 162 others voted for this review

Useful 31 | Funny 154 | Cool 18

References

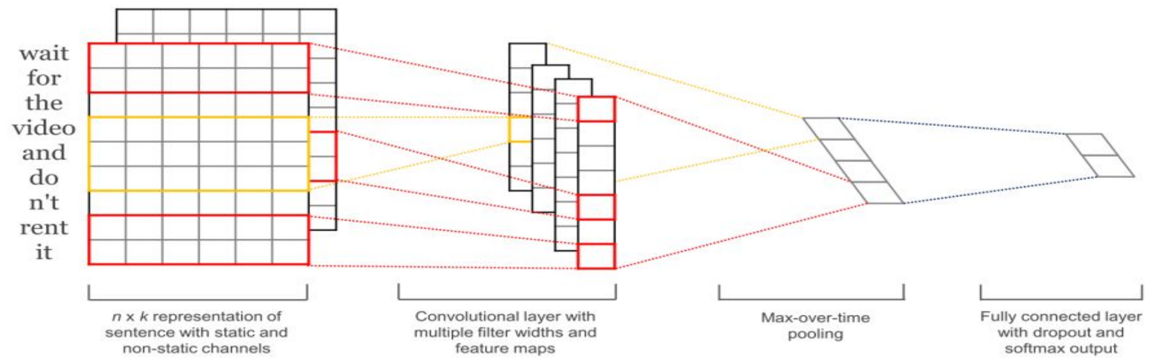
- L. de Oliveira and A. L. Rodrigo, "Humor detection in yelp reviews," 2015.
- Sosa, "Twitter sentiment analysis using combined lstm-cnn models," June 2017

Approaches

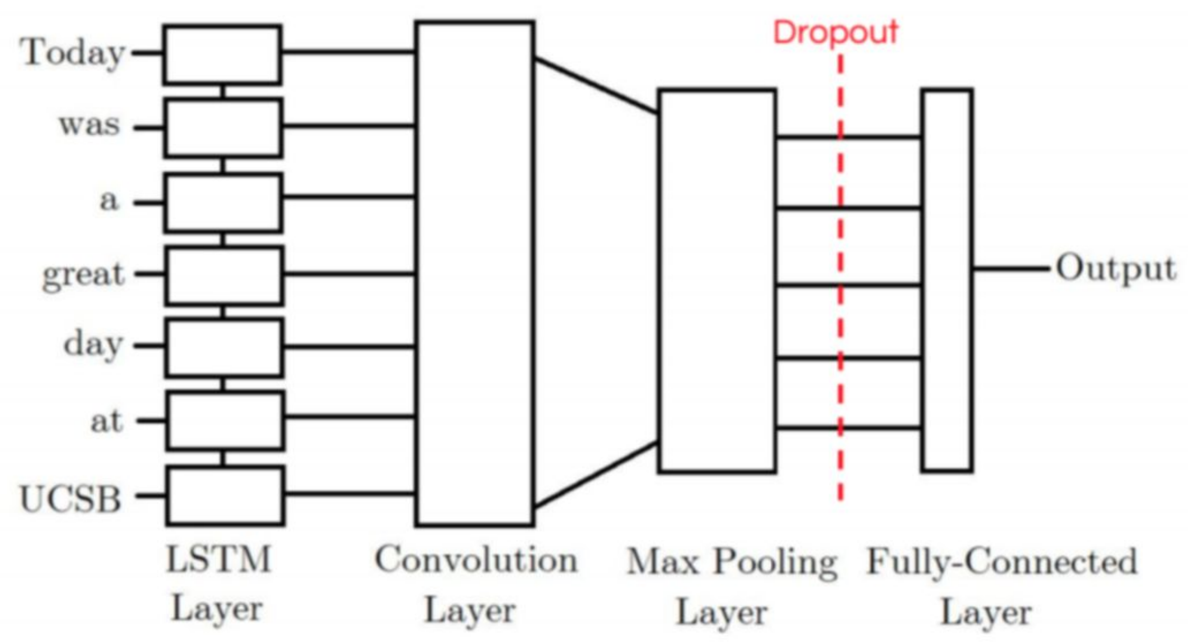
- **Baseline:** The baseline algorithm consisted of a 2-layer neural network that took in a sum of 256-dimensional word vectors as the input. Used extensions like tf-idf and removing punctuation.

$$tf-idf(w,d) = |w : w \in d| \log\left(\frac{N}{|w : w \in y, y \in D|}\right)$$

- **CNN:** Earlier method loses order information, but RNNs can be slow. Made a CNN that takes a sentence as a stack of word vectors and performs convolutions with different windows before max-pooling the results and feeding into fully connected layer.

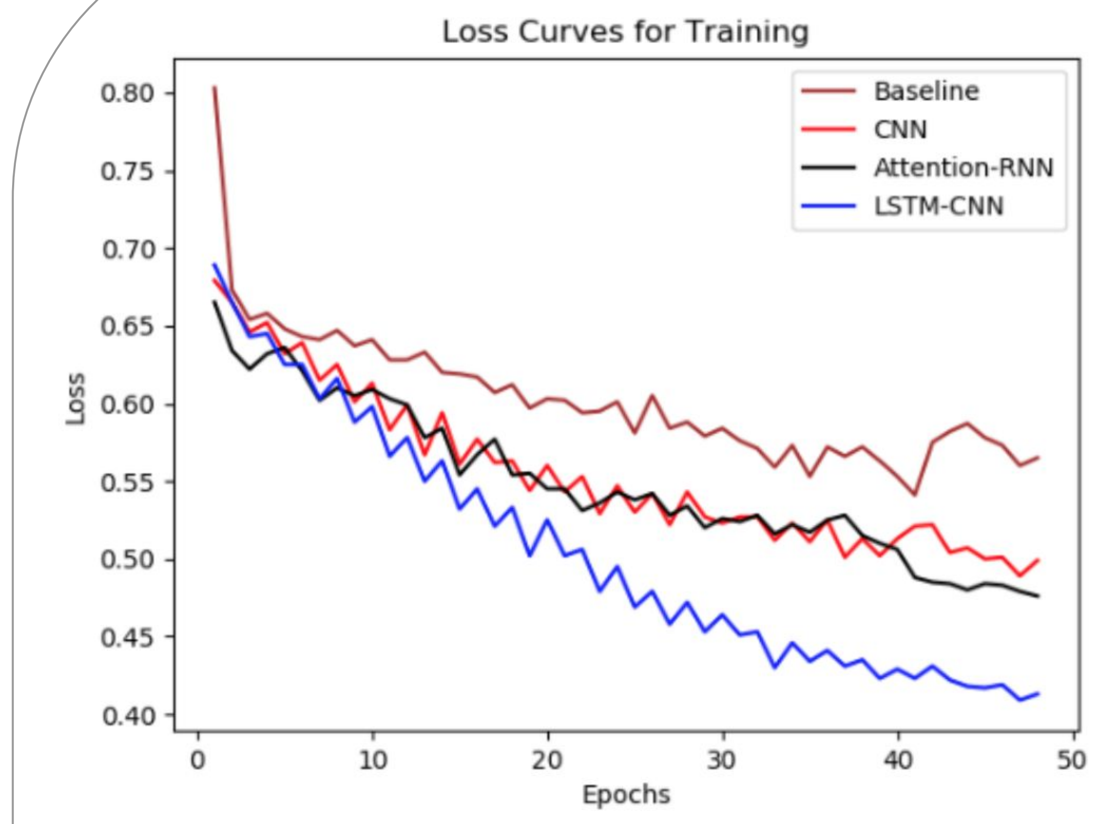

wait
for
the
video
and
do
n't
rent
it
n x k representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

- **Attention RNN:** Wanted to take advantage of sequential strengths of RNN but also modify the network to perform attention all hidden outputs and pass through softmax to give weightings to the hidden outputs to add together and pass to fully connected layer.
- **LSTM-CNN:** Pass embeddings to LSTM network to obtain hidden outputs and run previous CNN on top of the stacked matrix of hidden outputs.



LSTM-CNN in Sosa's paper.

Results



	Precision	Recall	F1-score
Baseline	0.602	0.603	0.602
CNN	0.653	0.622	0.637
RNN+attention	0.664	0.657	0.660
LSTM-CNN	0.723	0.698	0.711

- Evaluated with accuracy, precision, recall, F1
- The baseline returned about a 0.72/0.56 train/test accuracy.. Variations such as using TF-IDF, and treating punctuation as separate embeddings yields 0.78/0.62 train/test accuracy
- Approximately 0.8/0.6 train/test accuracy except LSTM-CNN had 0.9/0.7 train/test accuracy.
- F1 increased in order of Baseline, CNN, RNN, LSTM-CNN

Analysis

- TF-IDF improved the baseline, so the important words did indicate humor. Punctuation improved the baseline as well, since it helped show overreaction and transitions (! and ,).
- Severe overfitting because of high test accuracy/low test accuracy. Caused because humor in training set was direct story-based, and was sarcasm in test
- Example: "I looooove eating sushi in dingy apts."
- LSTM-CNN performed best with sarcasm (huge number of parameters to fit to model well enough)

Conclusion

- Sarcasm served as hurdle, causing some overfitting. "Important" words can indicate humor with TF-IDF
- Need to fix sarcasm issue with CNN-LSTM + attention