Stanford University
CS 221
Artificial Intelligence

React

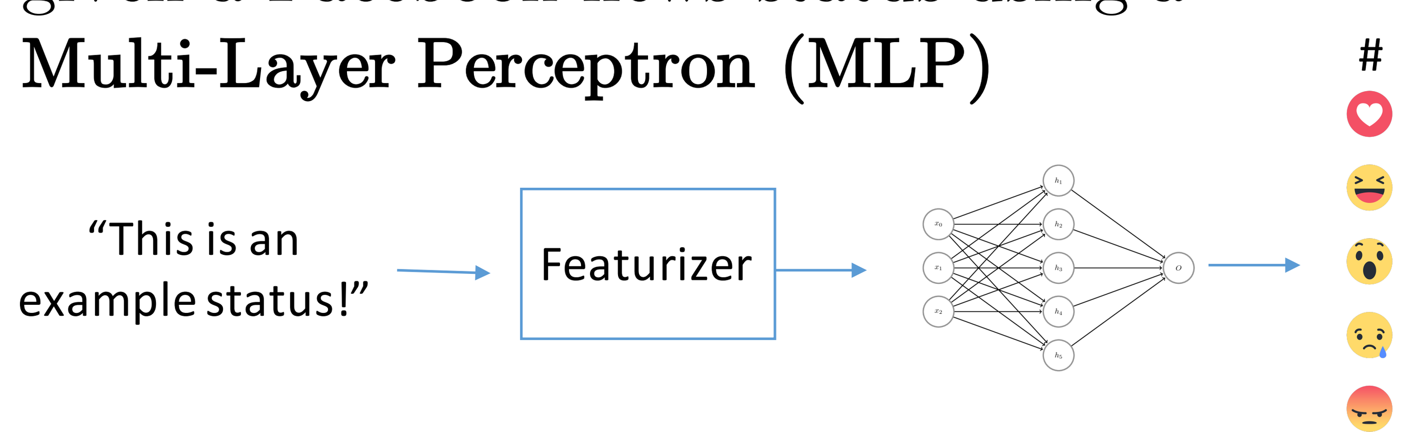
Derek Huang, Eli Pugh, Brandon Truong

Introduction

- Too much content, too many things going on in the world on Facebook to see all of it.
- Most of the time, you do not want to see certain moods or types of content, specifically types of news.
- It's also hard to know how people will react to your posts.
- Facebook Reactions provide rich information in how posts are received.
- It would be nice to:
    - Have an idea of how people will react to your posts (as a news site)
    - Be able to have filter and have guidance in what sort of content is being served based on Facebook Reactions.

Goal

Develop a multi-class prediction model that can infer proportions of Facebook Reactions given a Facebook news status using a **Multi-Layer Perceptron (MLP)**



Challenges

- **Context-dependence of texts:**
    - News statuses are time-sensitive and often cater to specific audiences.
- **Data aggregation:**
    - Majority of statuses have sparse reaction data.
    - Relative to other types of data, statuses are not that high in volume.

Scraped **13183** news statuses from 2017-2018 from high-traffic public-facing news pages, spanning political and social spectrum.

Approaches

Model:

- Multiple Layer Perceptron
- Squared loss
- Adam Optimization
- tf-idf

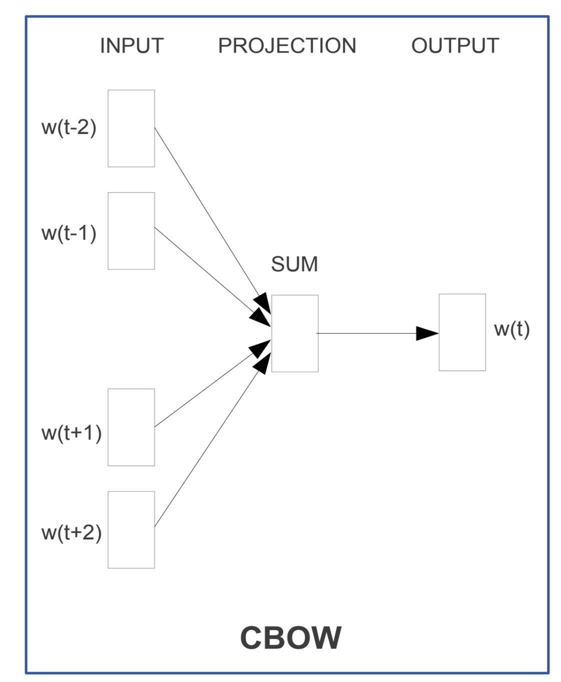
$$TF(t) = \frac{\#\text{of times term } t \text{ appears in a status}}{\text{total } \# \text{ of terms in the status}}$$

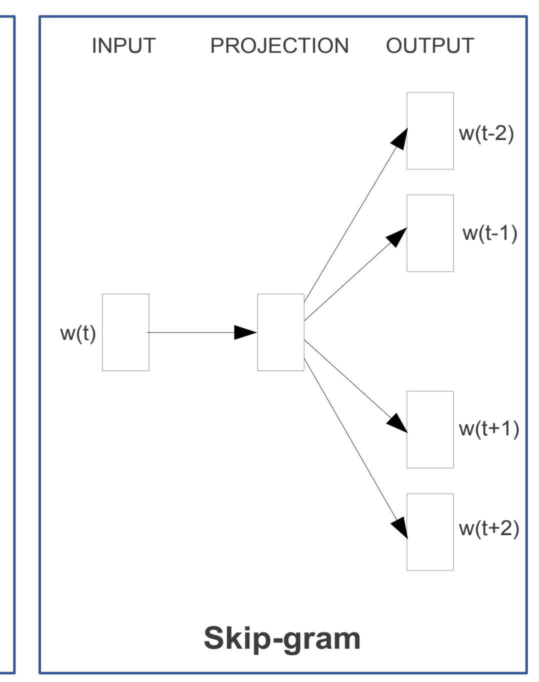
$$IDF(t) = \log_e \left(\frac{\text{total } \# \text{ of statuses}}{\# \text{ of statuses with the term } t \text{ in it}} \right)$$

Featurizers:

Our main area of focus was experimenting with different ways represent text to best extract sentiment. To be consistent, we tested all of our featurizers on the same model.

- Bag of Words - Unigram / Bigram
  Each post is represented as a sparse vector that counts the number of occurrences of a words and pairs of words
- Word2Vec
    - Continuous Bag of Words
      Learn to predict w_i from w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2}
    - Skip-Gram
      Learn to predict w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2} from w_i
- GloVe
    - Pre-trained word vectors
    - Word to word co-occurrence
    - Weighted least-squares objective





Examples

Example status 1

Police give an update on the 5 deaths at a nursing home in Florida after Irma http://cbsn.ws/2fiM0A6

Example status 2

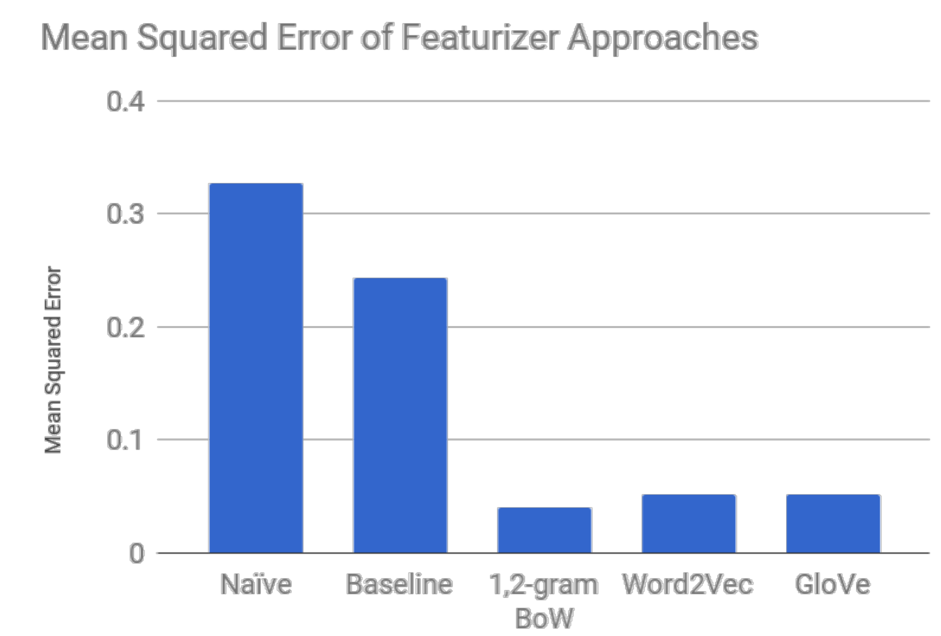
Algorithms are sets of rules used to help drive active decision-making. And they're lurking behind nearly every aspect of financial life. http://on.wsj.com/2roc1oo

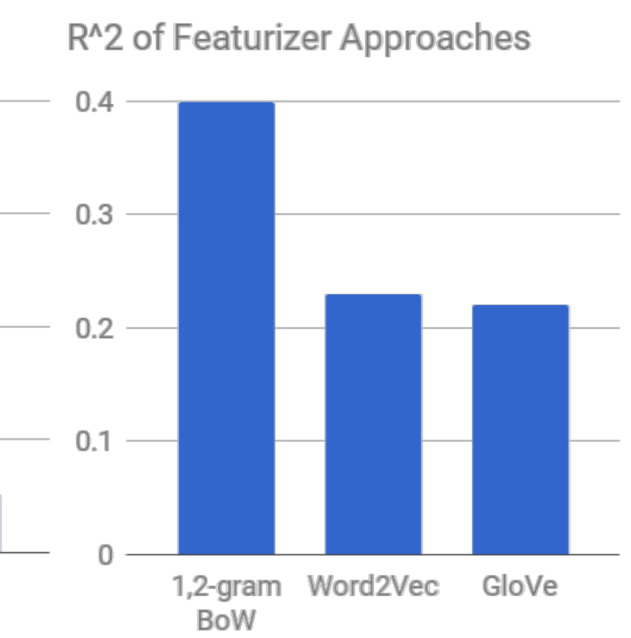
Predicted Output Reactions

Approach	❤️	😆	😮	😢	😡	❤️	😆	😮	😢	😡
Real	43	1	77	936	379	16	25	75	33	5
BoW	55	77	65	915	322	18	25	77	30	4
Word2Vec	0	105	304	930	96	25	23	53	37	16
GloVe	365	0	85	803	181	19	23	47	46	20

Results

- Surprisingly, BoW had best test error
- Improved 8-fold on Naïve (same prediction for all statuses)
- Improved 6-fold on Baseline (linear regression)





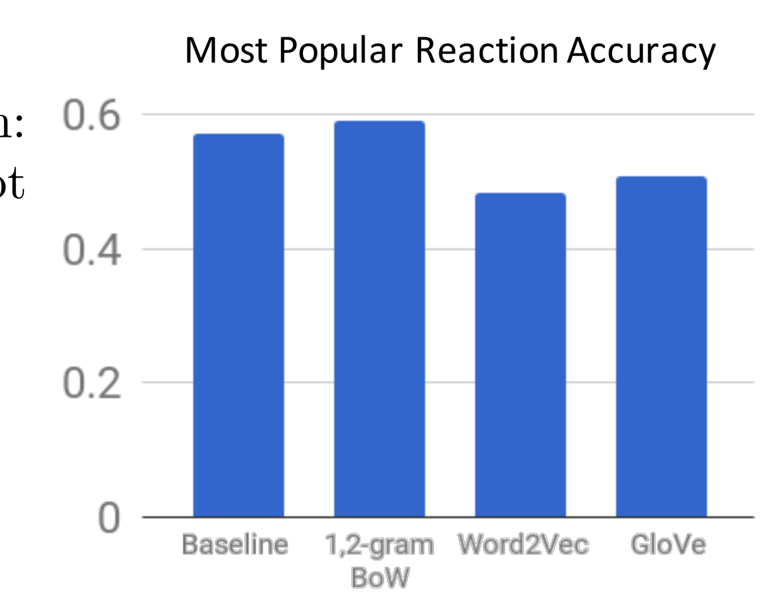
Analysis

Quantitative:

- Though BoW was best, with more training data Word2Vec and GloVe would improve
- The data was complex enough that we might have been more successful with Word2Vec and GloVe with an RNN

Qualitative:

- Upon inspection, our models perform worse with:
    - Very specific topics not seen often in training
    - Politically charged posts
- % correct of most popular reaction
- Outliers with only a few reactions skew scores



Moving Forward

Possible ways to improve:

- Scrape more training data from a wider variety of sources
- Use combinations of features
- Experiment with deep learning techniques such as RNN's