Analyzing Political Bias Through A User-Friendly Interface

Colin Lightfoot
Thesis Defense

Friday, December 8, 2017

Overview

1. Background
2. Related Work
3. Introduction to Neural Networks
4. LSTM Nodes
5. Recurrent Neural Networks
6. Implementation
7. Evaluation
8. References

Problem

1. Many news outlets report stories with biased undertones.
2. Some hide facts to present a specific argument.
3. The average person lacks the time to see multi-faceted stories.

Problem

1. Many news outlets report stories with biased undertones.
2. Some hide facts to present a specific argument.
3. The average person lacks the time to see multi-faceted stories.

Problem

1. Many news outlets report stories with biased undertones.
2. Some hide facts to present a specific argument.
3. The average person lacks the time to see multi-faceted stories.

Solution

1. Researchers have begun to use neural networks for textual analyses.
2. Recent studies have found neural networks to classify text accurately.
3. Use proven neural network techniques to create a useful political classifier.

Solution

1. Researchers have begun to use neural networks for textual analyses.
2. Recent studies have found neural networks to classify text accurately.
3. Use proven neural network techniques to create a useful political classifier.

Solution

1. Researchers have begun to use neural networks for textual analyses.
2. Recent studies have found neural networks to classify text accurately.
3. Use proven neural network techniques to create a useful political classifier.

Dataset

The Ideological Books Corpus (IBC):

- is a collection of sentences on multiple socio-political issues by recent US presidential candidates.
- Ex. "Liberal,Forcing middleclass workers to bear a greater share of the cost of government weakens their support for needed investments and stirs resentment toward those who depend on public services the most"

Related Work

- Iyyer et al. created political bias classifiers using Recursive Neural Network (RNN) models with Long Short Term Memory (LSTM) nodes with an accuracy of 69.3%.[1]
- Stanford researchers Arkajyoti Misra and Sanjib Basak showed that RNNs with LSTM nodes can also predict implicit political bias. Received an F1 Score of 0.718.[2]
- Both teams found RNNs with LSTM nodes are the most accurate models to perform textual analysis.
- Both teams found the IBC dataset as the most accurate dataset for political bias.

Neural Network Jargon

- Node: an object that stores a value (appears as a circle)
- Edge: a line connecting two nodes
- Epoch: an iteration/generation of a neural network
- Activation Function: a function which outputs a value once a threshold is reached

Introduction to Neural Networks

- Algorithms modeled after the brain.
- Organized into layers with interconnected nodes.
- Weighted edges connect nodes.
- Layers are successively computed based off computations from the lower layers.
- Nodes transform input using an activation function and then outputs result.

Introduction to Neural Networks

- Algorithms modeled after the brain.
- Organized into layers with interconnected nodes.
- Weighted edges connect nodes.
- Layers are successively computed based off computations from the lower layers.
- Nodes transform input using an activation function and then outputs result.

Introduction to Neural Networks

- Algorithms modeled after the brain.
- Organized into layers with interconnected nodes.
- Weighted edges connect nodes.
- Layers are successively computed based off computations from the lower layers.
- Nodes transform input using an activation function and then outputs result.

Introduction to Neural Networks

- Algorithms modeled after the brain.
- Organized into layers with interconnected nodes.
- Weighted edges connect nodes.
- Layers are successively computed based off computations from the lower layers.
- Nodes transform input using an activation function and then outputs result.

Introduction to Neural Networks

- Algorithms modeled after the brain.
- Organized into layers with interconnected nodes.
- Weighted edges connect nodes.
- Layers are successively computed based off computations from the lower layers.
- Nodes transform input using an activation function and then outputs result.

LSTM Nodes

- Retains standard node properties.
- Prevents exploding and vanishing gradient dilemmas during edges' weight training.
- Has a chance to "forget" its current state.
- Otherwise remembers state for next recurrence.

LSTM Nodes

- Retains standard node properties.
- Prevents exploding and vanishing gradient dilemmas during edges' weight training.
- Has a chance to "forget" its current state.
- Otherwise remembers state for next recurrence.

LSTM Nodes

- Retains standard node properties.
- Prevents exploding and vanishing gradient dilemmas during edges' weight training.
- Has a chance to "forget" its current state.
- Otherwise remembers state for next recurrence.

LSTM Nodes

- Retains standard node properties.
- Prevents exploding and vanishing gradient dilemmas during edges' weight training.
- Has a chance to "forget" its current state.
- Otherwise remembers state for next recurrence.

Recurrent Neural Networks

- Designed to make use of sequential information.
- Perform same computations for every element in a sequence.
- The output is dependent on the model's previous recurrences.
- RNNs can have dependent output since nodes hold previous calculations.

Recurrent Neural Networks

- Designed to make use of sequential information.
- Perform same computations for every element in a sequence.
- The output is dependent on the model's previous recurrences.
- RNNs can have dependent output since nodes hold previous calculations.

Recurrent Neural Networks

- Designed to make use of sequential information.
- Perform same computations for every element in a sequence.
- The output is dependent on the model's previous recurrences.
- RNNs can have dependent output since nodes hold previous calculations.

Recurrent Neural Networks

- Designed to make use of sequential information.
- Perform same computations for every element in a sequence.
- The output is dependent on the model's previous recurrences.
- RNNs can have dependent output since nodes hold previous calculations.

Why Not Markov Models?

- RNNs do not require the Markov assumption (current state is conditionally independent of previous states given the state in the previous timestep).
- Should not assume Markov assumption if there are other variables involved.
- RNNs are 'easier' to work with since there are numerous libraries implementing them for general use.
- Inefficiently scale in time the more states S are used at an asymptotic rate of $O(|S|^2)$.
- Table keeping track of probability of moving between states is of size $|S|^2$.
- Computation becomes impractical with large datasets

Why not other models?

- Support Vector Machines and Naïve Bayes bag-of-words models have been found to create less accurate models for textual analysis: classifies based off word frequencies than context.
- Convolutional Neural Networks (CNNs) have a hard time applying knowledge to unlearned situations: cannot grasp contexts of texts.

Implementation

Implementation

- User searches a news topic.

The user interface for my project.

Implementation

- Web scraper searches queried article results.

Implementation

- Articles are run through RNN.
- Cycle repeats until RNN finds both a liberal and a conservative article.

Implementation

Cross-Validation

- Only 67% of the dataset was used to train the model
- 33% of the dataset was tested against the model, creating the accuracies discussed in the oncoming slides
- the same 33% test data was split up to further test if model was well-trained but not over-fitting the 67% of the dataset on which it was trained

Accuracy

Accuracy is the percentage correct:

- Accurate prediction if model classifies sample with sample's attributed bias
- Overall Accuracy:

$$\frac{total\ test\ samples\ classified\ accurately}{all\ test\ samples}$$

- Conservative Accuracy:

$$\frac{total\ conservative\ test\ samples\ classified\ accurately}{all\ conservative\ test\ samples}$$

- Liberal Accuracy:

$$\frac{total\ liberal\ test\ samples\ classified\ accurately}{all\ liberal\ test\ samples}$$

F1 Score

Precision is measured as

$$\frac{true\ positives}{false\ positives + true\ positives},$$

while recall is measured as

$$\frac{true\ positives}{false\ negatives + true\ positives}.$$

Measuring the amount of false positives and false negatives shows model's ability to not predict Type I and II errors.
The formula for calculating the F1 Score is:

$$F1\ Score = 2 \times \frac{precision \times recall}{precision + recall}.$$

Evaluation

- F1 Score: 0.90
- Overall Accuracy: 57.49%
- Conservative Accuracy: 65.625%
- Liberal Accuracy: 57.065%

Evaluation

- F1 Score: 0.90
- Overall Accuracy: 57.49%
- Conservative Accuracy: 65.625%
- Liberal Accuracy: 57.065%

Evaluation

- F1 Score: 0.90
- Overall Accuracy: 57.49%
- Conservative Accuracy: 65.625%
- Liberal Accuracy: 57.065%

Evaluation

- F1 Score: 0.90
- Overall Accuracy: 57.49%
- Conservative Accuracy: 65.625%
- Liberal Accuracy: 57.065%

Limitations of Model

Machine Learning algorithms are trained by adjusting hyper-parameters to yield the most accurate models.

- The number of epochs was set by monitoring the improvement after each epoch.
- The amount of words remembered was set to 5000.
- The model forgot its values 20% of the time both before and after each round of learning.

Possible Noise

Possible Noise:

- Various-sized validation sets
- Complexity of the English language
- Samples with small sentence sizes

Future Work

- Continue tuning RNN
- Collect larger datasets
- Scrape more news outlets
- Run against other algorithms
- Better user interface
- Allowing for user feedback

Conclusion

Thank you, Professors Deverick and Lewis, as well as my friends and family for helping me complete this project! I was able to apply knowledge learned throughout my time here to create a user-friendly webpage that informs people about the political atmosphere surrounding a news event.

References

- Mohit Iyyer, Peter Enns, Jordan Boyd-Graber, Philip Resnik, 2014. *Political Ideology Detection Using Recursive Neural Networks*
  http://www.aclweb.org/anthology/P/P14/P14-1105.pdf
- Arkajyoti Misra and Sanjib Basak, 2016. *Political Bias Analysis*
  https://cs224d.stanford.edu/reports/MisraBasak.pdf
- Additional references in thesis.
  Link to this thesis' Github repository:
  https://github.com/hydure/HonorsProject

RNN In-depth

prediction
selection
collected
possibilities
forgetting
memory
possibilities

Activation Function

- softmax activation function for final layer:

$$softmax(n) = \frac{e^n}{\sum e^n}.$$

- softmax activation function outputs a value between 0 and 1 inclusive, which is the range the RNN needs to classify political bias
- other layers use the activation function shown below

RNN Mathematics

- x_t is the input at time step t
- s_t is the hidden state at time step t
- o_t is the output at time step t
- Data is stored in the hidden states s_t of a RNN and is calculated based on the previous hidden state and the input at the current time step:

$$s_t = f(Ux_t + Ws_{t-1}).$$

LSTM Mathematics

- input gate i_t at time t: $i_t = \varphi(W_i x_t + U_i y_{t-1} + b_i)$.
- forget gate activation function f_t at time t: $f_t = \varphi(W_f x_t + U_f y_{t-1} + b_f)$.
- output gate at time t: $o_t = \varphi(W_o x_t + U_o y_{t-1} + V_o C_t + b_o)$.
- Candidate values $\widetilde{C}$ readjust the state C_t at time t: $\widetilde{C} = tanh(W_c x_t + U_f y_{t-1} + b_f)$.
- new state C_t at time t becomes: $C_t = i_t \times \widetilde{C}_t + f_t \times C_{t-1}$.
- output y_t at time t: $y_t = o_t \times tanh(C_t)$.