Bias Detection in Sports Articles Using Structured Data
By Sherman Ng (SUNET ID: sqng629)
Email: sqng629@gmail.com, sqng629@stanford.edu
Link to video of the presentation on Youtube: https://youtu.be/goqVY_VhjP8

Overview and Problem Statement
Bias detection in news articles is a very difficult task where much research is currently being done. There have been quite a few different attempts at creating language models using machine learning and deep learning techniques such as linear discriminant analysis, recurrent neural networks, and LSTM. This experiment attempts to provide a solution to bias detection not by using language modeling approaches, but rather a methodology focused around text analysis. The methodology being tested is centered around first extracting text features, such as the text complexity and the number of words of certain parts of speech using tools such as the Stanford POS tags, and then passing the structured input into some deep learning network architecture to classify the article.

Dataset and experiment setup
This experiment specifically focuses on bias detection in sports news articles, and the dataset used is a database from the UCI Machine Learning Repository [1] containing 1600 named sports articles where part of speech and other text features such as the frequency of pronouns or question marks in the article, the frequency of proper nouns, and the number of foreign words used in the article are extracted. Each article in the database contains 59 total extracted features, so the full input dataset to the algorithms developed in this experiment can be viewed as a 1600 by 59 matrix. 90% of the dataset is used to train the network, while 10% of the remaining data is used as a validation dataset to measure the effectiveness of the trained network.

Simple Neural Network
The first model that I experimented with is a simple neural network comprised of 2 fully connected hidden layers, non-linear activation functions at each of the output of the hidden layers, an output layer, and a sigmoid activation function at the output. An Adam optimizer was chosen for optimization of the weights of the neural network, and backpropagation was used for optimization of the weight update. For the biases, L2 and L1 penalty are used, and stochastic gradient descent is used for training. The sigmoid activation function at the output of hidden layers instead produced a better validation accuracy rate. Adding more than 2 hidden layers did not seem to improve validation classification accuracy while increasing the training time to train the network, so the number of hidden layers used was kept at 2.

Convolutional Neural Network Architecture
Another model that I experimented with in an attempt to create an architecture that potentially generalizes better to unseen examples is a two-layer convolutional neural network. The 59 input features for each training example are passed as inputs into a network block whose first layer is a 1-dimensional convolutional layer, with kernel size and the number of filters determined empirically, in the first convolutional layer: for each training example, 10 filters of size 7 are convolved with the input features using “same” padding to make sure that input features on the edges do not get dropped out and in order to the number of filters in the convolutional layer and kernel size respectively. The rationale behind using an approach like this is based upon the hypothesis that certain input features may be correlated with each other. Using a CNN approach may enable the model to better capture the interdependencies between the features.

Results
Possibly because of the small number of training examples available for use in training the models, there is quite a bit of fluctuation in the validation dataset accuracy each time a model is trained and evaluated. To get a clearer picture of how a model performs when a certain subset of hyperparameters is used, a model is trained and evaluated 5 times for each subset of hyperparameters. For each model, a random 5% of the dataset is selected as the validation set while the remaining 95% is used for training. The performance of the model under a set of hyperparameters is then averaged over the 10 trials.

Since this experiment attempts to classify inputs into a binary output, a binary cross-entropy loss function was used to evaluate the performance of both models:

\[
\text{Loss} = -\sum \left( y \log(y') + (1 - y) \log(1 - y') \right)
\]

Simple Neural Network
Hyperparameters tuned:
- Number of layers in the network (2)
- Dropout keep probability (0.6-0.9)
- Number of Hidden nodes in each layer (300 in the 1st, 200 in the 2nd)

Convolutional Neural Network
Hyperparameters tuned:
- Number of convolutional kernel filters (5)
- Kernel size (2)
- Number of convolutional network blocks (2)

Regularization and Data Augmentation
Artificially created input feature = original feature + (rand(0, 0.2))

Conclusion
From the results of this experiment, there does not appear to be clear cut evidence of the superiority of either the simple neural network architecture or the convolutional neural network architecture in determining bias in sports news articles. Optimizing hyperparameters for either model also did not significantly improve classification accuracy. However, from the data augmentation experiment, it appears that perhaps by either artificially generating or collecting more training data to train the models, the convolutional neural network architecture may be able to generate and classify unseen articles better. This provides hope that given more time it is easier to improve data augmentation techniques and gather and collect more training data so that these models can be improved to better detect bias in news articles and systems as input can be developed and trained for higher classification accuracy. The difficult problem of bias detection in news articles can thus essentially be broken down into 2 steps, first by extracting text features from news articles and then by passing extracted features into the aforementioned models.

References: