Name: Qingyi Zhao
Email: qingyi@cs.ucla.edu

Detecting Political Bias in News Articles Using Headline Attention

Introduction

Major media press tend to hold attitudes for or against specific political parties, which are reflected in their news articles. As part of the general audience, we often find it difficult to distinguish which press and articles are biased while reading news articles. In the political domain, the media bias can range from selectively highlighting some events and leaders to focusing on facts advantageous for a particular party while ignoring those might hurt the public image of that party. For example, an article might purposely leave out certain context and make ambiguous assumptions to mislead the public audience.

It is very costly to manually debiasing a news article in terms of time, effort, and the amount of background knowledge required, but most of the time, we still want an unbiased analysis of a major incident such as presidential debate and candidate speech in important states. One solution to this problem is to use Natural Language Processing(NLP) techniques to automate the process of detecting bias in news articles. However, this task is very challenging due to the special nature of politically-biased news articles. Such biased articles are always written with extra care and attention from the authors since they want them to be biased in a subtle way such that it is sufficient to influence the mindset of the readers while not so obviously biased which will have the opposite effect.

Motivation

An interesting but sometimes ignored fact about reading those articles is that the readers usually read the headline first, and then finish the body of the article with that headline embedded in their mind throughout the whole process, which gives the headline of an article significant importance in terms of implanting biased terms and phrases. Apart from the headline, certain words in the body can be counted as keywords that have a larger weight to affect the objectivity of the reader than some other words. Therefore, it might be more beneficial to dig into the headline of an article as well as those keywords rather than focusing on every word with similar weight.

Background

This problem can be formulated as a text classification problem, where given a text, in this case a headline plus the article document, we want to classify it into one of the categories of our interests. Specifically in the political domain, one possible way to formulate is to treat political parties as different categories and classify a given article as biased towards one of the parties.

Word Embeddings

A common way to represent the words in natural languages in a way that machines can "understand" and process them is to use a technique called word embeddings. Simply put, given a sequence of words, we want to embed those words in a different format, usually in vector form or simply real numbers. The intuition is that words are in a high dimension where it is difficult to work in directly, and thus we want to reduce them to a low dimension vector space, or feature space where only related features of words are processed to reduce computational complexity. Traditional methods include dimension-reduction on co-occurrence matrices, probabilistic models, etc. But more recently, neural-network based approaches such as encoder-decoder or just a multi-layer perceptron have taken the lead and become the most popular choice in doing this task.

Text classification

Generally speaking, text classification problems can be split into two steps: extracting useful features and maximizing the classification margin based one those features. Some popular choice of classifiers with their choice of features include:

Naive Bayes classifier

In the most basic setting, naive bayes classifiers assume independence of each word or word features and take a probabilistic approach to classification with TFIDF score of each word as a common choice of features. It might not be as accurate for some tasks due to the Bag-of-Words assumption and context-independence assumption, but it achieved high performance in early days particularly for spam detection.

Support Vector Machine (SVM) classifier

SVM-based classifiers take an optimization approach with the objective of minimizing a max-margin or hinge loss and has proved its strength in domains such as text classification as well as image classification. Specifically in the NLP domain, some common choices of features include TFIDF score and N-gram models.

Neural Network methods

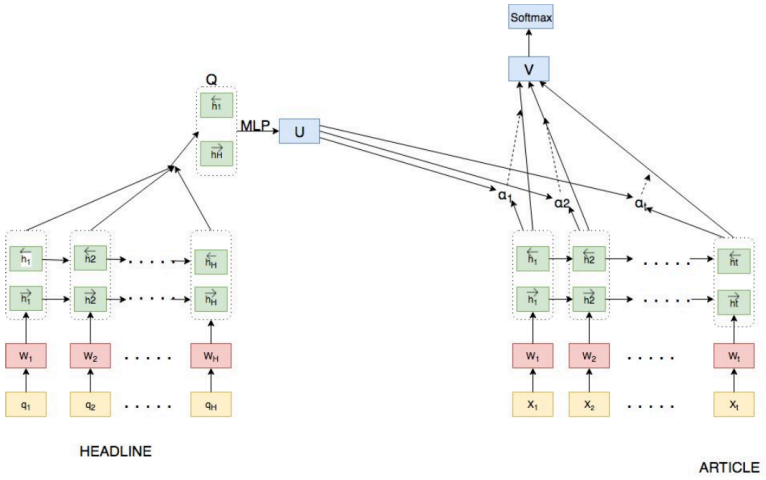
NN methods gained significance in recent years along with the development of deep learning. One major advantage is that recurrent architectures are claimed to capture the temporal information in a sentence or a text document which is difficult to collect with other traditional approach. Given the fact that understanding the semantic or pragmatics of a specific word often requires specifying the context that word or an entire sentence is in,

which sometimes is built on top of information provided at the very beginning of the document. Therefore, it is essential to record the history as well as the temporal identifiers for the correct context-dependent semantic of a word. Examples of such networks include LSTM and GRU. Later, some architectures, such as BERT, are proposed to also capture the information in the future sentences. We will leave the details out for the current discussion.

Model

Back to political bias detection, we formulate the problem formally as follows. Assume an article has T words, with w_i representing the i-th word in the article where i ∈ [1, T]. Similarly, assume there are H words in the headline, with q_i representing the i-th word in the headline where i ∈ [1, H].

The authors present a novel neural network architecture named Headline Attention Networks (HAN), which consists of several parts: a headline encoder, an article encoder, and a headline attention layer. The entire architecture is described in the below image:



Headline Encoder

$$x_i = W_e q_i, i \in [1, H]$$

$$\overrightarrow{h_i} = \overrightarrow{LSTM}(x_i) i \in [1, H]$$

$$\overleftarrow{h_i} = \overleftarrow{LSTM}(x_i) i \in [H, 1]$$

Given the headline of an article with words, q_i (i ∈ [1, H]), the authors first embed all words into vectors using an embedding matrix W_e : $x_i = W_e . q_i$, then use LSTM to generate a hidden representation, h_i for each word. An important note here is that two LSTMs are used: one for forward embedding and the other for backward embedding, with similar intuition as BERT that the context obtained from forward and backward directions can be enclosed in the

generated hidden embedding. Therefore the headline of an article will be represented as a tuple of two vectors: $Q=[\overleftarrow{h_1}, \overrightarrow{h_H}]$, where the two vectors are concatenation of the forward vectors $\overrightarrow{h_i}$ and backward vectors $\overleftarrow{h_i}$, respectively.

Article Encoder

$$x_i = W_e w_i, i \in [1, T]$$

$$\overrightarrow{h_i} = \overrightarrow{LSTM}(x_i), i \in [1, T]$$
$$\overleftarrow{h_i} = \overleftarrow{LSTM}(x_i), i \in [T, 1]$$

The Article encoder has essentially the same motivation with the Headline Encoder, by first embedding the words into a lower-dimensional embedding using traditional word embeddings techniques, and then use two LSTMs from two opposite directions to incorporate bi-directional contextual information. The only difference with the Headline Encoder is that the resulting encodings for Article Encoder are per word, i.e. the annotation of word w_i is represented as $h_i=[\overrightarrow{h_i}, \overleftarrow{h_i}]$.

Headline Attention Layer

This layer is an additional layer essentially designed for identifying keywords in the article that represent similar attitude as specified in the headline. Before any input is passed to this layer, the headline embedding is again embedded into another representation U. Then each hidden word representation h_i is compared to the hidden headline embedding to compute a similarity score which is then used to compute the bias of each word towards a specific category.

The hidden representation of each word h_i is first transformed into u_i, another layer of embedding, as

$$u_i = tanh(W_w h_i + b_w)$$

,

which is dotted with the hidden headline matrix U in the softmax-like way as

$$\alpha_i = \frac{exp(u_i^T . U)}{\sum_i exp(u_i^T . U)}$$

And the hidden representation of the amount of bias can be expressed as a weighted sum of all words based on their importance weights:

$$v = \sum_i \alpha_i h_i$$

.

Output

The final prediction is given as the output of the final layer of the entire HAN architecture with softmax applied:

$$p = Softmax(W_c v + b_c)$$

The loss used to train the network is the negative log-likelihood of the correct labels; in other words, how much data can the prediction explain the ground-truth assignment:

$$L = -\sum_d Log(p_{di})$$

where i is the label of document d.

Results

The dataset used for in this paper is created and annotated by the authors, and data sources are various Telugu newspapers. There are 1329 new articles in total in the dataset, all annotated manually by the authors themselves. This step might have already introduced bias in the ground-truth data labels, but let's ignore that possibility for now for the sake of this discussion.

The dataset is divided into three parts for experiments: headlines only, articles only, and both. The authors have measured performance of the proposed with multiple baselines including Naive Bayes and SVM-based approaches as well as deep learning methods including traditional CNNs, branched CNNs, and LSTM/GRU. All experiments are done using 5-fold cross-validation on the datasets. Results are summarized in the table below:

Methods	Only Headline	Only article	Concatenation of headline and article	Maximum
Naive Bayes+TFIDF+Unigrams	39	58	59	59
Naive Bayes+TFIDF+Bigrams	29	32	33	33
Naive Bayes+Bag-of-means	49	63	63	63
SVM+TFIDF+Unigrams	41	69	69	69
SVM+TFIDF+Bigrams	55	76	71	76
SVM+AverageSG	57	69	66	69
CNNs	80	80.5	81.7	81.7
Branched CNNs	83.33	84.52	84.6	84.6
LSTM	84	85.25	85.32	85.32
GRU	81	82.7	82.7	82.7
Headline Attention Network without attention layer	-	-	-	**85.25**
Headline Attention Network	-	-	-	**89.54**

From the table, we can see that the proposed model outperforms the previous best (LSTM) by 4.22%. An interesting fact to note is that the simple HAN without the attention layer has similar performance as LSTM, which justifies the importance of the attention layer, i.e. the keywords in the article body that are similar to the headline impose larger influence to the readers in terms of the intended bias they carry.

A potential problem with the experiments is that all experiments are done on a self-annotated dataset. To make the results more persuasive, it will be better to apply the proposed method to some more popular and stable benchmark for political bias datasets.

Conclusion

The authors proposed a novel neural network architecture to detect political bias in news articles. The problem itself is formulated as a multi-class classification problem. The main contribution of the HAN architecture is the introduction of headline attention: extracting keywords from the body of the article based on the similarity between each word and the headline, with the intuition that the headline should impose the most amount of bias from the perspective of the authors of the biased articles. The proposed model outperformed the state-of-the-art model on a dataset created an annotated by the original authors of this paper.

Related Works

1. Mohit Iyyer, Peter Enns, Jordan Boyd-Graber, Philip Resnik, “Political Ideology Detection Using Recursive Neural Networks”, ACL 2014.

   This paper claims to be the first to apply recurrent neural networks for identifying political ideology detection. The existing methods mostly assume the bag-of-word assumption and thus are lack of contextual information. By using RNN, the authors incorporate hierarchical structure of the text into consideration and outperformed existing models. Another contribution is that this paper introduced a new way to crowdsource ideological bias annotation in the domain of political data.

2. Lazaridou, Konstantina. “Identifying Political Bias in News Articles.” TCDL Bulletin 12 (2016): n. Pag.

   This paper essentially has the same goal as the surveyed paper above. However, the focus is a bit different. It introduces a more structured classification of different types of political biases which can all be categorized as political *slants*. They include, but not limited to, choice of media outlets, range or degree of an event being covered, the way facts are presented. The authors didn’t propose any models per se, but focused the experiments on the counts certain entities such as Politician’s mentions and Politicians’ quotations. The results were reported on datasets Guardian and Telegraph by doing phrase queries with Elasticsearch.

3. Holtzman, N.S., Schott, J.P., Jones, M.N. et al. “Exploring media bias with semantic analysis tools: validation of the Contrast Analysis of Semantic Similarity (CASS) ”Behav Res (2011) 43: 193.

   This paper presents a software tool called CASS to compare association between groups and within a model. Given various media articles that associate liberal or conservative with bad or good terms, the authors report the result of CASS tested on these inputs. In other words, they tested on whether CASS is able to detect the political attitude of media press from their published articles based on the sentiment words those articles used towards a political party. From my understanding, what CASS does is compute the semantic similarity of two words based on a special metric implanted in this piece of software.

4. Ceren Budak, Sharad Goel, and Justin M. Rao “Fair and Balanced? Quantifying Media Bias through Crowdsourced Content Analysis ”Public Opinion Quarterly, Vol. 80, Special Issue, 2016, pp. 250–271

   This paper investigates the selecting and framing of political issues of 15 major US news outlets, and found out that political bias mostly occurs in articles on political scandals. Other than that, most of them exhibit fair presentation of a political event, casting neither Democrats nor Republican. Moreover, except for political scandals, they also post no major difference in story selection. They built two classifiers using large-scale logistic regression to identify bias in a particular article. The models are trained against crowdsourced labels of 749 online human judges.

5. Daniel Xiaodan Zhou, Paul Resnick, Qiaozhu Mei “Classifying the Political Leaning of News Articles and Users from User Votes ”Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media, 2011

   This paper presents three semi-supervised learning methods to propagate political leaning of known articles and users to the target unknown users or articles. It is not really bias detection per se, but it is useful in the sense that if a model can clearly classify an article into a certain political party, then it already shows that the article is biased towards that party. The results showed a huge improvement in the performance compared to the traditional SVM-based approach. We could compare the model with some recent deep learning methods on the same dataset to see the performance as a possible future direction.

Reference

Gangula, R.R., Duggenpudi, S.R., & Mamidi, R. “Detecting Political Bias in News Articles Using Headline Attention”. ACL 2019.