CSE 472: Social Media Mining

Project II

Prof. Huan Liu
TA: Zhen Tan

Understanding the Impact of Ideological Bias on NLP Tasks

Hirthik Mathavan (1225184012)

Trishal Gayam (1225073096)

Abstract

This project experiments with different Topic classification models on Ideologically biased datasets. The dataset is collected from news tweets on twitter with segregation of Left, Right and Neutral sided content based on AllSides Media Bias Chart. Our experiments prove that classification algorithms show the effect of bias and recognize the direction of future improvements for specific Natural Language Processing (NLP) tasks such as Topic classification.

Introduction

Political bias in the media is a complex issue which has an impact on the behaviour of humans. In today's world, many news media sources support specific political parties and their ideologies. The news from the article reflects the author's view and understanding as well as the media source organization. There are two primary ideologies in political parties, namely conservative ideology and liberal ideology. Conservatives (Right) believe that government should be small, operating mainly at the state or local level. They favour minimal government interference in the economy and prefer private sector-based solutions to problems. "Social conservatives" believe that government should uphold traditional morality and impose restrictions on contraception, abortion, and same-sex marriage. The definition of liberalism (Left) has changed over time. However, modern-day liberals tend to believe that the government should intervene in the economy and provide a broad range of social services to ensure well-being and equality. Liberals usually believe that the government should not regulate private sexual or social behaviours.

Twitter is one of the significant sources of information spread. Every major news media has an account on Twitter where they tweet their news articles around different categories: politics, world, us, sports, entertainment, etc. The open debate supported through messages and comments on the news tweets attracts more people. More exposure to biased information causes ideological segregation and can affect voting behaviour. This makes it more interesting to understand the bias in the news information spread on Twitter through different news sources categorized into (left, right, and neutral) based on AllSides.

In this project, we focus on identifying the effect of Ideological bias in news data from Twitter on different topic classification models. Many topic classification tasks depend on breaking down the text and understanding the patterns through syntactic and semantic structures. A robust model should also mitigate the effect of content-specific bias in the training and test data. Finally, we would be able to understand the reason for the research question: do these AI models suffer from ideological bias too?

Related Works

- We Can Detect Your Bias: Predicting the Political Ideology of News Articles - In this paper, the authors have explored the ideological bias in news articles incorporating Twitter bios (medium followers) and Wikipedia (medium content) and categorizing the dataset topics into elections, immigration, coronavirus, and politics. They evaluated LSTM and BERT models and found improvements through Triplet Loss Pre-training (TLP).
- Algorithmic Amplification of Politics on Twitter - This research was done for twitter's home timeline personalization algorithm. The algorithm which ranks the content on our personal home feed making some content more visible. It had a side effect of amplifying specific political groups because of the ideological bias present in them.
- News Sharing User Behaviour on Twitter: A Comprehensive Data Collection of News Articles and Social Interactions - This dataset consists of information about some news articles that were shared by normal/regular users to identify dissemination of news information. This dataset is also used in our experiments.

Model Description

Datasets:

- We haven't done any additional preprocessing of the dataset like removing stop words, converting words to lowercase, removing special characters from text, etc.
- We chose 'politics' and 'world' as out labels as the text content for both these classes is similar.

Dataset 1 - (Twitter News) - We scraped twitter tweet data for 3 news sources FoxNews (Right), CNN (Left), and The Wall Street Journal (Neutral). We collected 740 news tweets for each topic: politics and world from every news media twitter account, that accounted for a total of 4440 news tweets. We distributed 1000 news tweets (500 'politics' and 500 'world') as our training set for one side of a news source and 480 news tweets as our test set for each news source separately.

Process:-

- Used 'snscrape' library to search tweets over a period of date range with the twitter user id for news media (FoxNews, CNN, NewsNation)
- Every tweet had text content of the news headline with the tiny url attached with the tweet.

- We extracted headlines as our text data. Expanded the tiny url to the actual url path and extracted labels through string matching keywords (‘/politics/’ and ‘/world/’) as the news articles are grouped based on the topic and have specific url paths.
- This way we had Ground Truth Labels without any manual intervention.
- The data we collected was based on recent months published articles May-September months.
- The pre_processed dataset also has the news urls linked with the tweet content.

Dataset 1 - (Harvard News) - We downloaded the Harvard dataset of news articles shared on twitter and processed the article content with the same topics: politics and world from 3 news sources FoxNews (Right), CNN (Left), and NewsNation (Neutral). The same number of text-label pairs are gathered for the Harvard dataset similar to the Twitter tweet dataset above.

Process:-

- The Harvard dataset of articles had specific structure with topic labels and news article content.
- The text used here is actual news data.
- The news articles are shared by different users on Twitter.

Model Descriptions:

- We have used some general models like Naive Bayes, SVM (Support Vector Machines), Decision Tree, Random Forest, and KNN (K- Nearest Neighbors).
- We also used complex classification models FastText and BERT for our experiments.
- All the general models we used methods (svm, naive_baise, KNeighborsClassifier, tree, and RandomForestClassifier) from the sklearn library to train and test the data.
- Used CountVectorizer and TfidfTransformer for tokenization and vector representation of the words in the text in general models.
- For SVM we used a linear kernel with regularization parameter 1.0.
- For the Random Forest model we used the n_estimator parameter (for number of trees) as 100.
- For Naive Bayes we used the MultinomialNB method. **MultinomialNB** implements the naive Bayes method for multinomially distributed data.
- sklearn.metrics had methods for accuracy_score and f1_score to calculate our model performance measures.

Special Models:

FastText:

FastText, by Facebook Research, is a library for efficient learning of word representations and text classification. FastText supports supervised (classifications) and unsupervised (embedding) representations of words and sentences. For the classification task, multinomial logistic regression is used, where the sentence/document vector corresponds to the features.

- For our experiments we ran the FastText model for 1000 epochs.

BERT:
Bidirectional Encoder Representations from Transformers is known by the abbreviation BERT. Transformer encoders are organized in layers and make up the BERT architecture. Each Transformer encoder also has two sub-layers: a feed-forward layer and a self-attention layer. Utilizing a Transformer, BERT can identify the relationships between words in a sentence or text based on their context. An encoder that reads the text input and a decoder that produces a prediction for every given task are two independent mechanisms included in the transformer. In order to produce a language model, BERT solely uses the encoder.

- For our experiments we ran the model for 10 epochs with a batch size of 32.
- We used adam optimizer with a dropout of 0.1.

Experiment

Setup

- We trained models separately on 3 sides of the dataset (1000 data rows) and tested each model on 3 sides of the dataset (480 data rows).

F1 - Score table

Model\Bias	Dataset	Left	Right	Neutral
Naive Bayes	Twitter	Left - 0.93817 Right - 0.78899 Neutral - 0.82679	Left - 0.80537 Right - 0.94021 Neutral - 0.83197	Left - 0.6581 Right - 0.88211 Neutral - 0.89655
	Harvard	Left - 0.93362 Right - 0.91868 Neutral - 0.87665	Left - 0.83532 Right - 0.93246 Neutral - 0.87336	Left - 0.88435 Right - 0.89732 Neutral - 0.89076
SVM	Twitter	Left - 0.95178 Right - 0.81739 Neutral - 0.82483	Left - 0.83197 Right - 0.92784 Neutral - 0.86667	Left - 0.79018 Right - 0.87767 Neutral - 0.930328
	Harvard	Left - 0.94694 Right - 0.94672 Neutral - 0.89754	Left - 0.92614 Right - 0.96721 Neutral - 0.87896	Left - 0.93548 Right - 0.94456 Neutral - 0.92339
KNN	Twitter	Left - 0.93252 Right - 0.80078 Neutral - 0.82955	Left - 0.7907 Right - 0.88372 Neutral - 82.305	Left - 0.74523 Right - 0.82483 Neutral - 0.89571
	Harvard	Left - 0.87671	Left - 0.80919	Left - 0.77052

		Right - 0.87594 Neutral - 0.8629	Right - 0.86081 Neutral - 0.84109	Right - 0.75974 Neutral - 0.83365
Random Forest	Twitter	Left - 0.91213 Right - 0.77419 Neutral - 0.81922	Left - 0.85653 Right - 0.91031 Neutral - 0.8254	Left - 0.78532 Right - 0.81532 Neutral - 0.87083
	Harvard	Left - 0.92562 Right - 0.92784 Neutral - 0.87026	Left - 0.862 Right - 0.94958 Neutral - 0.68633	Left - 0.91295 Right - 0.92735 Neutral - 0.88
Decision Tree	Twitter	Left - 0.87064 Right - 0.75862 Neutral - 0.81481	Left - 0.73782 Right - 0.85520 Neutral - 0.69307	Left - 0.73179 Right - 0.74365 Neutral - 0.84232
	Harvard	Left - 0.86192 Right - 0.82526 Neutral - 0.82377	Left - 0.82887 Right - 0.85532 Neutral - 0.79295	Left - 0.82083 Right - 0.80493 Neutral - 0.79079
FastText	Twitter	Left - 0.90417 Right - 0.77292 Neutral - 0.77292	Left - 0.78125 Right - 0.9 Neutral - 0.83542	Left - 0.73958 Right - 0.83958 Neutral - 0.89167
	Harvard	Left - 0.86042 Right - 0.86667 Neutral - 0.77917	Left - 0.82708 Right - 0.90625 Neutral - 0.78958	Left - 0.83125 Right - 0.87083 Neutral - 0.84792
BERT	Twitter	Left - 0.81758 Right - 0.47337 Neutral - 0.42168	Left - 0.6217 Right - 0.83983 Neutral - 0.80543	Left - 0.33229 Right - 0.78082 Neutral - 0.67708
	Harvard	Left - 0.8535 Right - 0.84116 Neutral - 0.77261	Left - 0.77966 Right - 0.8415 Neutral - 0.78459	Left - 0.79078 Right - 0.83685 Neutral - 0.78354

Outcomes & Observations

- Difference between Twitter and Harvard Data
    - The classification models executed on two types of text headline of news article (Twitter) and actual news article data (Harvard).
    - All the classification models didn't generate any major difference in the F1-Scores on Harvard dataset. The small changes might be possible because of the author's language used in different media sources and differences in news

articles published on the sources.
    - For Twitter dataset every model in general had higher F1 score for the same train and test set even when the data contained closely related articles headlines (closely related days news on politics and world).
    - **Conclusion:-**
        - In the Harvard dataset because of more text context size the classifiers were easily able to differentiate between politics and world news even with bias present in the data.
        - The headlines on Twitter data were more targeted with the bias and due to less information the classification models had represented the effect of bias. As the author's view is less effective on small headline statements.
- From all these models SVM gave the best performance on average of 3 sides of training and testing for both Twitter dataset and Harvard Dataset.
- The BERT model gave less F1 scores in general for the datasets as we trained only for 10 epochs on the data. We could recognize the model overfit the data when trained on a Left biased set that gave very low F1 scores when tested on Right and Neutral.
- (Focus on Twitter Data) In general models trained on Neutral dataset should have higher F1-scores on average for testing on 3 sides or the difference of F1-scores between test sets should be close, compared to Left and Right side training. If we consider our model is performing better.
    - **Naive Bayes** - The Neutral Training data is more leaning towards Right side because of close F1-Scores and big difference with Left side test data
    - **SVM** - This model also had similar closeness of Netral Training data with Right side test set but able to better classify in all 3 test sets than other models
    - **KNN** - This model also had similar closeness of Netral Training data with the Right side test set.
    - **Random Forest** - This model also had similar closeness of Netral Training data with the Right side test set.
    - **Decision Tree** - This model also had similar closeness of Netral Training data with the Right side test set.
    - **FastText** - This model also had similar closeness of Netral Training data with the Right side test set.
    - **BERT** - Strangely the model had higher accuracy score for Right side test set than the Neutral set and proves the Bias had the highest effect on the BERT model.
    - **Conclusion** -
        - The effect of bias was more on Naive Bayes and BERT models.
        - The SVM model worked best.
- (Focus on Twitter Data) In general, models trained on Left data should have higher F1 score difference between Left and Right test data compared to Left and Neutral and Right training dataset should have higher F1 score difference between Right and Left test data

compared to Right and Neutral dataset if our classification model is affected by Ideological bias.

- **Naive Bayes** - Model works similar to our assumption
- **SVM** - Model works similar to our assumption
- **KNN** - Model works similar to our assumption
- **Random Forest** - When trained on Right dataset was able to classify Left test set with more F1-Score than Neutral set. In our Random Forest classifier we use n_estimators parameter as 100 and as the numbers of trees increase it increases the robustness of the model. Have verified with different settings (1, 10, 100, 1000) of that parameter but identified the same pattern.
- **Decision Tree** - Model works similar to our assumption
- **FastText** - Model works similar to our assumption
- **BERT** - Has huge difference of F1 scores for trained set and test set groups. Shows BERT is highly affected by Ideological bias.
- **Conclusion** -
  - BERT model has highest effect of Ideological bias
  - Random forest algorithm can be explored deeply with respect to Ideological bias as it shows different behavior towards general classification as compared to other models.

Final Conclusions:

- BERT model uses its custom word embeddings that internally use LSTM model to build word vectors and capture the context.
- FastText uses a bag of n-grams to identify the information of the words. Also it is very fast in training the model (trained for 1000 epochs) within a few seconds on our dataset compared to BERT. It doesn't have much effect of Ideological bias in classification and follows with the general models.
- Other models used basic vectorization for the words using CountVectorizer and TfIdfVectorizer from sklearn library and had less effect of Bias on the models compared to BERT.
- **Our Question - do these AI models suffer from ideological bias too?**
  - Yes, from the results the model shows some representation of bias.
  - The BERT model which captures more relation between words present in the data and had the highest effect of all.
- **How can we make our models more robust?**
  - Topic classification worked well in general vectorization of word relations.
  - As we build into more domain specific classification models there will be a requirement of capturing more information based on the relation of words in the text. This maximizes the criteria to improve the models when presented with biased datasets.

Future Works

- The major context of Ideological Bias is identified on the political spectrum. So working on classification models on sub-categories of political data would reveal better insights on the model performance.
- We haven't used other word embedding structures like Word2Vec, Glove, and ElMo, based on the time constraints. But, analyzing these models would also give some more understanding and learning. Especially models perform better when the word embeddings are generated on actual train and test content rather than the use of globally available datasets of word embeddings that have been modeled for the general case.
- We planned to try label classification by combining data from all three sides with 3 labels (left, right, and neutral) and experiment on a combined test dataset if the models are able to predict the articles correctly or identify if any new news article is not biased. This requires many code changes in file paths and also the model codes have to be generalized. So, we planned to keep it for future research scope and extend the work.

Additional References

- https://www.allsides.com/media-bias/media-bias-chart
- https://www.analyticsvidhya.com/blog/2021/12/text-classification-using-bert-and-tensorflow/
- https://medium.com/@bedigunjit/simple-guide-to-text-classification-nlp-using-svm-and-naive-bayes-with-python-421db3a72d34
- https://github.com/kk7nc/Text_Classification
- https://medium.com/text-classification-algorithms/text-classification-algorithms-a-survey-a215b7ab7e2d
- https://github.com/facebookresearch/fastText
- https://fasttext.cc/docs/en/supervised-tutorial.html
- https://www.khanacademy.org/humanities/us-government-and-civics/us-gov-american-political-ideologies-and-beliefs/us-gov-ideologies-of-political-parties/a/lesson-summary-ideologies-of-political-parties
- https://libguides.uwgb.edu/c.php?g=600365&p=4157309
- https://towardsdatascience.com/fasttext-bag-of-tricks-for-efficient-text-classification-513ba9e302e7#:~:text=FastText%20supports%20supervised%20(classifications)%20and,representations%20of%20words%20and%20sentences.
- https://aclanthology.org/E17-2068.pdf