Political ideology analysis using Natural Language Processing

Avanti Bhandarkar, Keerthiraj Nagaraj
Department of Electrical and Computer Engineering
University of Florida
avantibhandarkar@ufl.edu, k.nagaraj@ufl.edu

***Abstract*— In this project, we aim to employ principles of Natural Language Processing and sophisticated deep learning algorithms that can be trained to classify text sentences based on the ideology that it closely aligns with respect to US political parties. Natural Language Processing (NLP) is becoming increasingly popular in detecting private states such as opinions, sentiment, and beliefs, and we will be using this ability of NLP to classify text sentences based on political ideologies.**

I. Introduction

In this section, we will briefly introduce various technical concepts required to understand the analysis carried out in our project.

Short form	description
NLP	Natural Language Processing
IBC	Ideological Book Corpus data
TF-IDF	Term Frequency Inverse Document Frequency
SVM	Support Vector Machines
NBC	Naive Bayes Classifier
LogReg	Logistic Regression
RNN	Recurrent Neural Networks
LSTM	Long Short Term Memory units
GRU	Gated Recurrent Units
TITC	Training with IBC, Testing with convote
TCTI	Training with convote, Testing with IBc
SSL	Semi Supervised Learning

TABLE I: List of abbreviations used in this report

A. Foundations of NLP

With the manifold increase in data over the last few decades, it has become imperative to be able to automatically analyze the data and use the specific information gained, for a given purpose. Automatic information retrieval has been gaining momentum in all the fields. One such field which is getting a lot of attention lately, is Natural Language Processing (NLP), as it studies the limit to which computers can help humans understand natural human language, in applications like question-answering, speech recognition, machine translation and human machine communications like chat-bots etc. It is a wide area of research which includes amalgamation of Artificial Intelligence, Computer science as well as Linguistics.

B. Text classification

Text classification is a type of supervised machine learning method which is used to classify documents or sentences in one or more predefined categories. Text classification is used in applications like sentiment analysis, spam filtering, categorizing text samples in different topics etc. We will be using text classification techniques to classify 2 American political ideologies i.e. Liberal and Conservative, and hence our problem becomes a binary classification problem.

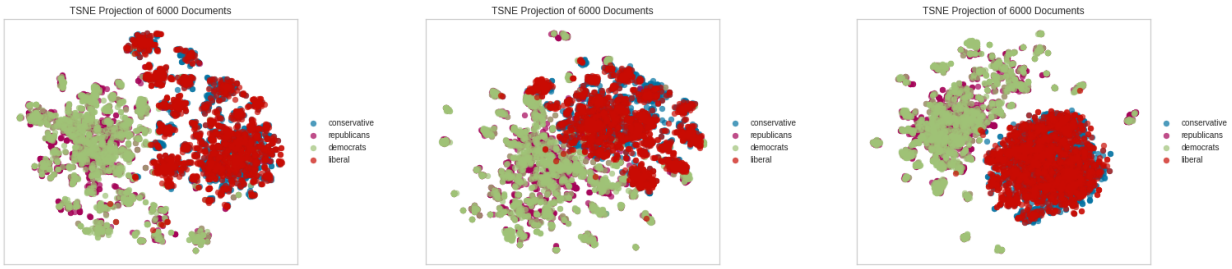
C. Background on political ideologies

The political atmosphere in the US is deeply polarized between two predominant ideologies: liberal and conservative. The two major parties maintain differences in opinion in different issues like Climate change, Gun laws, LGBTQ rights, Abortions, Foreign policies and Immigration, to name a few. While the liberals encourage active role for government in society and believe in environmental regulations, conservatives like to have a limited role for government in the society and argue against imposing environmental regulations.

Our hypothesis is that two major party system in USA has led to more polarized views between politicians towards major issues facing the country today. Politicians belonging to different parties have strong opposing views on most of the prominent issues and because of this the bipartisan support for any policy becomes difficult to achieve. This may not be the case in multi major party systems like the one India has, as politicians doest have to always strongly favor or oppose a policy as political bias spectrum is distributed more fairly compared to USA. Understanding the bias that the politicians from different parties have might help us reducing it and bringing in more politicians together to work towards bipartisan support for common good of the country.

D. Natural Language Processing for political ideologies

Detecting or recognizing an ideological bias is very difficult for humans, as it not only relies on the choice of words or political knowledge of the person, but also on subtle key differences in use of language. Politicians are so careful with their words, sentences, and phrasing etc, that even small change in the word choice can result in taking opposite ideological stand. We can often see that the parties supporting a certain bill/objective would use more glorified words to express their support, whereas the parties



(a) BOW features (b) TFIDF features (c) BOW+TFIDF features

Fig. 1: t-SNE visualization in feature space

against that bill/objective would make use of harsher or grim-sounding words/phrases to convey their opposition. One such example that is often subject of political debates, is that of Estate tax, which is a tax on property that gets transferred to a person upon someone elses death. As the Democratic party supports this tax, they call it Estate Tax or Inheritance tax, whereas Republican party, which does not support this tax call it Death Tax.

Identifying such subtleties can be of utmost importance while classifying ideological bias. These subtleties, even though difficult, can be picked up by humans. But in this age of big data, manually identifying ideological bias is impractical as well as extremely expensive process. Moreover, bias can be located at any position of the sentence, phrase or paragraph and hence we are unable to hard-code algorithms to detect this bias using computers. In this project, we will examine different machine learning algorithms to classify text into 2 ideologies.

E. Related work

There has been a lot of work in exploring if and how ideological bias or stance can be recognized using opinion analysis [1]. The increased interest in constructing political opinion classifiers, can be attributed to potential applications like e-Rulemaking or even to gauge the general public opinion.[3] Researchers have tried to classify opinions in domains such as Online newsgroup discussions (newsgroup20 dataset), congressional speeches and debates as well as based on written documents such as books and manifestos by politicians expounding their ideas. Mohit Iyyer et al, mention that a sentence contains ideological bias if the bias is evident, given the authors political position. An interesting example of ideological stance mentioned in Somasundaran et al. [1] is an individuals answer to Do you believe in God?. A person might argue for as well as against the existence of god. Ideological and political topics together are very popular as it not only involves the topics, but also a persons personal opinions.

Bias has not only been studies for political opinions and ideologies, but also to detect bias in journalism. Currently, every media outlet has their own take on current news, and often, certain media organizations post opinionated articles, which may further affect the opinions of the reader. Doumit et al, proposed a method based on Latent Dirichlet Allocation (LDA), where the researchers treat different media sources as separate classes with different personalities. They have made use of the topics as well as concepts to extract personality traits for each class. They also predict what inclination a class of media agent would take on certain topic or concept based on their model.

Various approaches have been used to detect and classify political bias in a sentence. We can often see that classification of ideologies cannot be a yes/no type binary problem. A person might have divided opinions when it comes to ideologies. This has been very efficiently dealt with in [4], where the researches used text obtained from twitter and experimented with language features like uni-grams, word clusters and emotions to study the differences between 2 ideologically extreme groups, 2 ideologically moderate groups and between both extremes and moderates, total of 7 ideological groups. For most of the work in this field, Bag of Words features have proved to work well so far, on a limited number of samples.

With the increase in amount of data, the rule-based approach and bag of words do not seem to capture all the required information, given the limited information capture of word frequencies by bag of words approach. Bag of words lacks the ability to pick up contextual information, which plays a very important role in language understanding. With the rise of Deep Learning models for applications such as self-driving cars, health care, voice activate assistants etc, it was an obvious choice for researchers to test these models on natural language. For Natural Language Processing, the sentences are broken down into words, which are considered tokens to the deep learning model. These models treat sentences as a sequence of tokens. [5] There is also a debate among linguists about the structure of language being hierarchical since characters together compose words, words together compose sentences and so on until we have multiple paragraphs which compose document. To deal with this hierarchical structure of language Socher et al, proposed a new architecture which treats the sentences in a hierarchical manner. Building on this concept, [7] authors have introduced a way to parse ideologies from a single sentence. We can observe that not all the words in a sentence convey the same ideology and a single word can completely flip the meaning of the sentence. They achieved a 70% classification accuracy as they can capture long term dependencies owing

to the deep learning models architecture. We will be using some concepts from all the previous works to analyze this problem of ideology detection.

II. Database description

A. Ideological Book Corpus Dataset

We are using a dataset by the name, Ideological Books Corpus (IBC), which was developed by a group of researchers from the University of Maryland. The data set was based on publicly available US congressional floor debate transcripts from 2005. The final data set was a compilation of sentences handpicked to be most expressive of political sentiment and manually labeled by a majority voting scheme. We are using the entire data set for this project that consists of 2025 liberal and 1701 conservative biased sentences.

B. Convote Dataset

Convote dataset consists of text sentences from US congressional floor debate transcripts collected during 2005 with each sentence labelled according to the party affiliation of the congressman. This dataset has been mainly used for document-level and sentence-level political party classification. There are 3711 sentences from Democrats and 3708 sentences from Republicans, a total of 7419 sentences. The assumption made for this dataset is that party affiliation is directly related to the ideological stance of sentences. Hence, all sentences belonging to congressmen/congresswomen from Democratic party have liberal ideology and all sentences spoken by representatives of Republican party are conservative in nature. We also aim to verify whether this assumption holds true with our experiments.

Statistic	IBC	Convote
Number of liberal sentences	2025	3711
Number of conservative sentences	1701	3708
Total number of words	85487	1016800
Total number of characters	628655	7330698
Average word length	5	5
Median words per sentence	22	42

TABLE II: Data statistics

C. Preprocessing

Pre-processing involves transforming the text data, to a form which is recognizable by the natural language processing algorithm. Our pre-processing function has 2 main parts:

- Cleaning: This consists of removing unwanted part of text, by methods like, 1. Stopword removal: removes the words that do not contribute towards the meaning of the sentence, 2. Removing capitalization : treating all words as lower case, so each word with same spelling is given equal weightage, 3. Removing symbols and punctuation: these do not play any part in classification of text.
- Normalization: This consists of translating words with different parts of speech to their root form, or generalized word form. For example, the word 'festival' can occur in the text as 'festive', 'festivity' etc. Converting the words to their root form, guarantees the occurrence of that word only once in the vocabulary of the data. Normalization is done using 2 methods 1. Stemming: chopping the endings of the words (coarse method), 2. Lemmatization: converting the words to their root form based on vocabulary and morphological analysis of words.

D. Feature extraction

1) Bag of Words: Bag of words is the simplest model used to extract features from text data, and has been a great success for tasks such as language modeling as well as text classification. Bag of words is just a count of words in the document. It consists of 2 things:

1) A dictionary or vocabulary of all the word appearing in the text
2) Number of times a particular word occurs in the document

The name "bag" of words, directly conveys that, any information about the sequence of words or structure of words is lost with this feature representation. with this feature, we can understand if a particular word occurs in a given document, but there is no way to know where in the document.

2) TF-IDF: The problem with using bag of words approach is that words with highest count will have higher scores. The most commonly occurring words may not have any important information and hence, we may get poor performance on our machine learning or deep learning model using just bag of words approach. For example, we may have the preposition "of" with highest counts in our document, and hence will be given the highest score. But in practice, a preposition would not add any information gain. To fix this problem, TF-IDF, which stands for term frequency inverse-document frequency, assumes that the most frequently occurring words are the one's with least information and penalizes the words that occur frequently across all documents. TF-IDF is used to analyze the importance of a word to the document in a collection of documents.

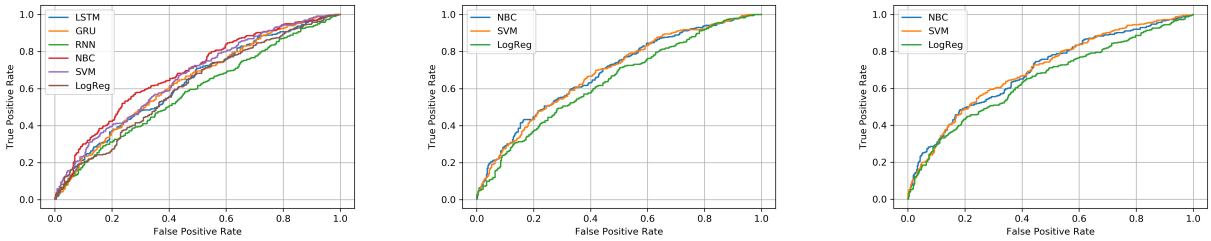
3) TF-IDF + BOW: BOW and TF-IDF are usually the first steps in text analytics pipeline. A bag of words feature representation create frequency of each words occurring in the document, whereas tf-idf gives more importance to the less occurring words while considering the common frequently occurring words redundant. Thus these two features together, are used to convert a raw text document to numerical feature representation.

III. Machine learning and Deep learning models

In this section, we discuss about the models that we have used to perform sentence classification in our project.

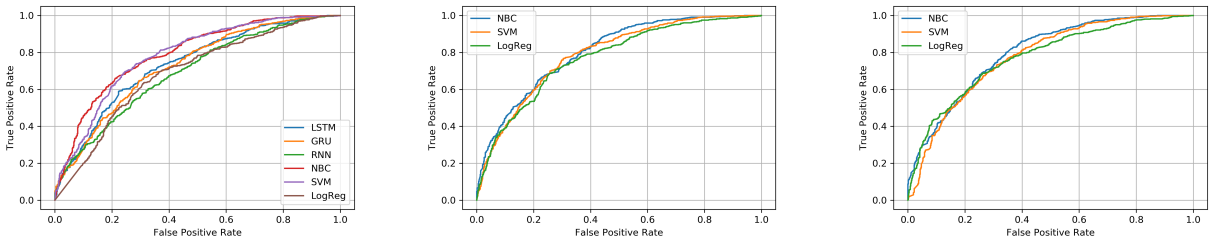
A. Logistic Regression

Logistic Regression is a statistical method for classifying a dataset with a set of independent input variables that determine an output with 2 classes. The objective of this



(a) BoW (b) TFIDF (c) BoW + TFIDF

Fig. 2: ROC curves for IBC dataset



(a) BoW (b) TFIDF (c) BoW + TFIDF

Fig. 3: ROC curves for Convote dataset

technique is to find the best fitting model that can describe the dichotomous relationship between input samples corresponding to the output values. This method makes use of logistic curve or also known a sigmoid curve that can efficiently separate samples from different classes in the dataset.

B. Naive Bayes Classifier

Naive Bayes Classifier is a classification technique developed based on the principles of Bayesian theorem and is highly suited when working with dataset containing large number of dimensions. Naive Bayes Classifier can work with arbitrary number of independent variables whether are categorical or continuous. For a given set of input values (training data), the idea is to construct a posterior probability for each output class, which is then used to measure class membership values for unseen data (test data).

C. Support Vector Machine Classifier

Support Vector Machines are supervised learning models mainly used for solving classification and regression problems. It is an example for non-probabilistic classifier. In classification problems, SVM tries to maximize width of the decision boundary gap that separates samples from two distinct classes. SVM can be used with both linear and non-linear classification problems. In our experiments, we have used SVM with 'hinge' loss function and 'l2' regularization.

D. Recurrent Neural Networks

Humans depend on the previous words to understand the context of next word in a sentence and it is important to use models that can capture this behavior while learning how to classify text sentences. Recurrent Neural Networks address this issue by having loops in its structure and hence allowing past information to persist during training.

RNN is a class of artificial neural network which can capture temporal information from the input data and use it while training the model. At each iteration, the function value of the output depends on the current input sequence and the hidden state which contains the information from past iterations.

E. Long Short Term Memory model

RNN can capture only capture short term dependencies in the data, and sometimes in order to understand the meaning and context of a given word, we need to have long term dependencies. Long Short Term Memory units (LSTM) are a special kind of RNNs which can help in that regard. LSTM also has chain like structure similar to RNN but the repeating part is slightly different as it contains 4 other functions interacting with each other instead of a single neural network layer. The 4 functions in the repeating module of LSTM are Point-wise Operation, Vector transfer, Concatenate and Copy.

F. Gated Recurrent Units

Gated Recurrent Units are another kind of Recurrent Neural Networks which can also capture long term dependencies but can do so with lesser number of parameters than LSTM. GRUs contains gating mechanism which contains Update gate, Reset gate, Current memory content and Final memory at current time step gate. With this gating mechanism, GRUs are able to store and filter information. GRUs also eliminates the vanishing gradient problem as the learning model keeps relevant information and passes it down to the next time steps

of the network without disregarding evry new input sample after that iteration.

Parameter name	Value
Recurrent layer	20
Dense layer	10,1
Activation function	Sigmoid
Loss function	Binary Cross entropy
Optimizer	ADAM
Validation patience	3 epochs
Dropout Ratio	0.3
Early Stopping	Yes
Number of epochs	10

TABLE III: Deep Learning architecture

IV. Experiment design and results

A. Experiment 1: t-SNE Visualization in feature space

t-distributed Stochastic Neighbor Embedding, or t-SNE, is one of the most popular methods for visualizing similarity between 2 documents. This is a machine learning algorithm for non-linear dimensionality reduction [8], similar to principal component analysis(PCA). In this experiment, we want to look at the relationship of vectors obtained from 3 different feature sets and compare these high-dimensional features in a 2-dimensional feature space. Using t-SNE, we expect that similar vectors will be modeled closer, whereas unlike objects will be modeled far apart in the reduced feature dimension. Similarity in the feature vector depends on the feature set being used, for example, for bag of words model, words which usually occur together will be modeled closer like for our dataset "political party" are 2 words which mostly occur together.

We randomly sampled 1500 samples from each of our classes from both the datasets and extracted Bag of Words (BOW), TFIDF as well as combination of BOW+TFIDF from the text and plotted the features using t-SNE algorithm into 2 dimensional feature space. We can see from the figure 1, that even though there is a clear separation between ideology and party affiliation i.e IBC dataset and Convote dataset, our assumption that liberal ideological features will be closer to democratic party samples, does not hold true. Furthermore, observing the spread of samples within a dataset(IBC-liberals vs conservative ideology, Convote-Democrats vs Repulican party affiliation), we can not see a distinct separation between classes. In the following experiments, we would like to corroborate this observation by training and testing these datasets on machine learning as well as deep learning models.

B. Experiment 2: Ideological Books Corpus Data

In this experiment, we trained and tested machine/deep learning models using IBC dataset. Our objective for this experiment was to develop models that could capture the political ideology from text sentences. We have repeated this experiment with 3 different features for 3 machine learning models and with Bag of Words feature for 3 deep learning models. For machine learning models, the highest test accuracy that we could obtain is 63.00% with Naive Bayes Classifier and Bag of Words feature, and for deep learning models, the highest test accuracy is 60.19% with Gated Recurrent Units.

Experiment	SVM	NBC	LogReg
BOW IBC	62.60	63.00	59.78
BOW Convote	72.23	69.74	65.97
BOW Overall data	68.28	68.10	62.09
BOW TCTI	54.50	57.59	50.94
BOW TITC	52.20	50.99	51.57
TFIDF IBC	61.79	61.26	59.51
TFIDF Convote	72.37	71.56	71.36
TFIDF Overall data	67.56	68.50	66.44
TFIDF TCTI	55.28	56.98	52.60
TFIDF TITC	51.30	50.81	51.21
BOW +TFIDF IBC	62.19	60.00	57.64
BOW +TFIDF Convote	70.61	68.80	70.75
BOW +TFIDF Overall data	67.56	68.50	66.44
BOW +TFIDF TCTI	55.28	56.98	52.60
BOW +TFIDF TITC	51.30	50.81	51.21

TABLE IV: Performance of Machine Learning models

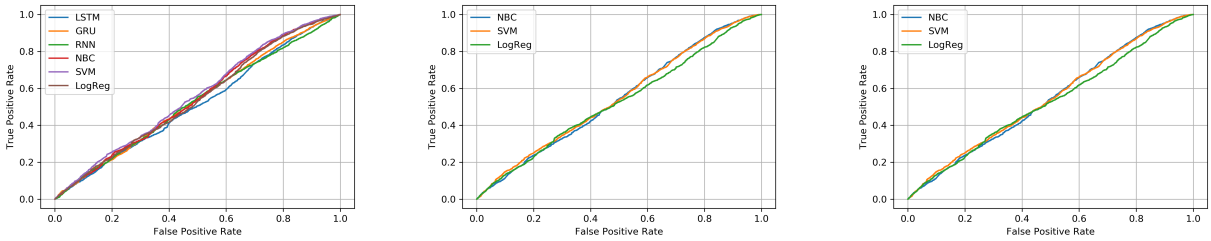
We believe the low accuracy particularly from deep learning models is due to the fact that IBC dataset has small number of input samples and even the class size is unbalanced with liberal biased sentences having much higher number of samples than conservative biased sentences. The labels were created by multiple humans with diverse viewpoints and so there is also a problem where the input samples might have not been labelled properly as each individual with different political ideology can look at same sentence with different perspectives. Ideology is a subjective topic, hence the labelling also impacts the classification bias.

Figure 2 shows the ROC curves for all the experiments conducted for IBC dataset with varied combinations of features and learning models.

C. Experiment 3: Convote Data

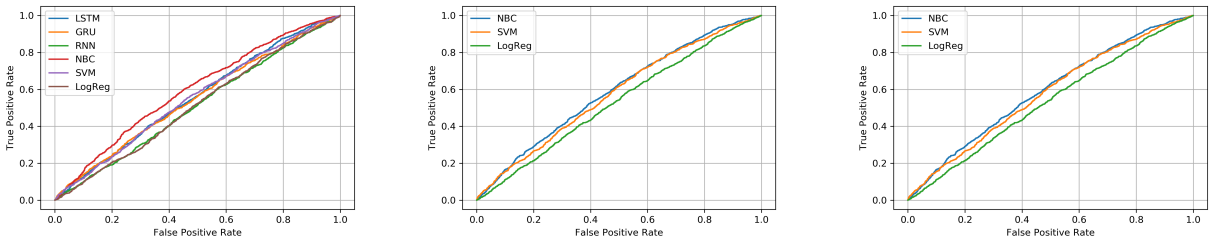
In this experiment, we trained and tested machine/deep learning models using Convote dataset. Our objective for this experiment was to develop models that could capture the political party from text sentences. We have repeated this experiment with 3 different features for 3 machine learning models and with Bag of Words feature for 3 deep learning models. For machine learning models, the highest test accuracy that we could obtain is 72.37% with Naive Bayes Classifier and TFIDF feature, and for deep learning models, the highest test accuracy is 67.79% with LSTMs, although GRU also results in similar performance with an accuracy of 67.25% on test dataset.

The performance improvement can be attributed to the fact that convote dataset has larger number of sentences compared to IBC dataset, the class sizes are also balanced here. This dataset was labelled based on the party affiliation of the speaker whose congressional speech transcript was used to



(a) BoW (b) TFIDF (c) BoW + TFIDF

Fig. 4: ROC curves for Train IBC Test Convote



(a) BoW (b) TFIDF (c) BoW + TFIDF

Fig. 5: ROC curves for Train Convote Test IBC

get a given input text sentence. We believed that it is easier to classify political party association of text sentences than the political ideology into which it falls as congressmen from same political party tends to agree on similar issues and hence more often than not, use similar language to express their positions on different topics. The better performance accuracy with convote dataset also aligns with this belief.

Figure 3 shows the ROC curves for all the experiments conducted for IBC dataset with varied combinations of features and learning models.

D. Experiment 4: Cross testing

In the literature, we noticed that Convote dataset was used for political ideology analysis but the dataset was labelled based on the party affiliation, which is based on the assumption that party affiliation and political ideology are very closely related. To test this assumption, we trained machine/deep learning models on one kind of dataset and tested its classification accuracy on the other.

We initially used IBC data for training and tested the model performance on convote dataset (TITC) and then trained a model using convote dataset while testing it with IBC data (TCTI). A good performance in these cross testing experiments would support the assumption that political ideology and political party affiliation has high correlation in these two datasets. For TITC, with machine learning models, the highest test accuracy that we could obtain is 52.20% with Support Vector Machines and Bag of Words feature, and for deep learning models, the highest test accuracy is 51.66% with Gated Recurrent Units. For TCTI, with machine learning models, the highest test accuracy that we could obtain is 57.79% with Naive Bayes Classifier and Bag of Words feature, and for deep learning models, the highest test accuracy is 54.48% with Gated Recurrent Units.

With TITC, training size is smaller than test size, so the model was not able to give good results. The best accuracy of 52% was almost equal to predicting the classes randomly. TCTI was able to give slightly better results because of higher training size compared to test size.

Figure 4 shows the ROC curves for TITC experiments and 5 shows the ROC curves for TITC experiments for all the 3 different features.

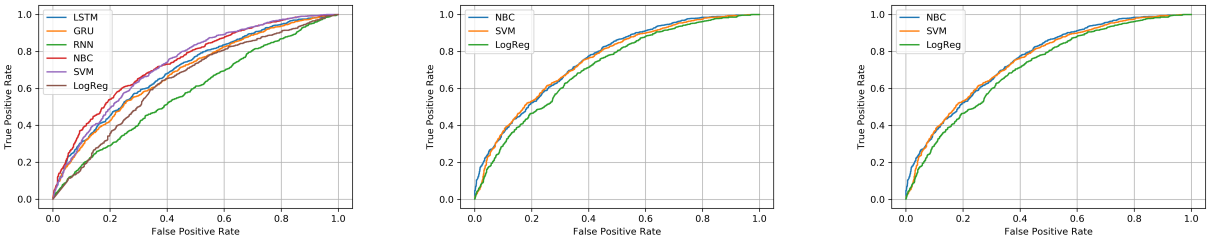
E. Experiment 5: Combined dataset

In this experiment, we trained and tested machine/deep learning models using samples from both IBC and convote dataset. We conducted this experiment to again the test the assumption that political ideology and party affiliation has high correlation as claimed in the literature. For machine learning models, the highest test accuracy that we could obtain is 66.44% with Logistic Regression and BoW+TFIDF feature, and for deep learning models, the highest test accuracy is 63.62% with LSTMs. The results that we obtained goes against this assumption as were not able to get good test accuracy.

Figure 6 shows the ROC curves for all the experiments conducted for combined dataset with varied combinations of features and learning models.

F. Experiment 6: Semi-supervised learning

Semi Supervised Learning (SSL) is mainly used when we have large unlabelled data and small amount of labelled data. We used labels from one dataset for the initial model training and pseudo labelling and retrained the model with initial



(a) BoW (b) TFIDF (c) BoW + TFIDF

Fig. 6: ROC curves for combined dataset

(a) SSL with IBC training (b) SSL with convote training

Fig. 7: Semi Supervised Learning

training set and pseudo labelled samples. Pseudo labelling refers to labelling of data samples in an unlabelled dataset using a model which was trained with some other dataset.

Firstly, we divided the IBC data into training and test datasets, we then trained a neural network based model with IBC training data and saved the model with best validation accuracy. We labelled the text sentences of Convote dataset using the saved model and called them as pseudo labels. We then trained a new neural network model with initial training samples and pseudo labelled samples combined, and tested its performance with IBC test dataset. Without semi-supervised learning, the best test accuracy for IBC with deep learning models was 60.19%, but with semi-supervised learning we were able to achieve higher accuracy of 65.01%.

Secondly, we divided the Convote data into training and test datasets, we then trained a neural network based model with convote training data and saved the model with best validation accuracy. We labelled the text sentences of Convote dataset using the saved model and called them as pseudo labels. We then trained a new neural network model with initial training samples and pseudo labelled samples combined, and test its performance with convote test dataset. Without semi-supervised learning, the best test accuracy for convote data with deep learning models was 67.79%, but with semi-supervised learning we were able to achieve higher accuracy of 68.94%.

Figure 7 shows the ROC curves for all the experiments conducted for Semi-supervised learning.

Test accuracy for all the experiments conducted in our project are mentioned in Table IV and Table V.

Experiment	LSTM	GRU	RNN
IBC	57.77	60.19	56.17
SSL with IBC	65.01	61.13	57.64
Convote	67.79	67.25	63.48
SSL with convote	68.94	68.13	66.78
Overall data	63.62	62.49	55.85
TCTI	54.48	54.37	51.15
TITC	50.73	51.66	50.98

TABLE V: Performance of deep learning models

V. CONCLUSIONS

In this paper, we have explored 2 widely used datasets for political ideology detection. IBC dataset is a compilation of liberal and conservative annotated sentences, based on US congressional floor debate transcripts, whereas Convote dataset consist of raw US congressional floor debate transcripts, with party affiliation annotations. Starting out, we made an assumption that given the source of both our datasets is the same, the ideological stance taken by one of the parties should remain consistent. We assumed that all the samples from Democrats belong to liberal ideology and all samples belonging to republicans have conservative ideology. Our goal was to prove that there is an evident relationship between party affiliation and ideology. To test our hypothesis, we selected 3 feature sets, namely, Bag of Words, TF-IDf and the combination of both. We wanted to see the performance of these features on machine learning as well as deep learning frameworks and hence we selected 3 models for each. In all of our experiments, we got the best performance on training and testing tfidf features on convote dataset. We attribute

this performance to the number of samples available in this dataset. We can argue that, we got a better performance on the same dataset, with only TFIDF features on a machine learning SVM classifier, with an accuracy of 72.37%, than the work by [5],[7] which uses word2vec features on an LSTM deep learning classifier and gets an accuracy of 70%. Although, the performance gain is not huge, the fact that we used a simpler feature representation and classifier, proves the generalization ability of our model. Finally, our assumption that political party affiliation is directly related to the ideological stance taken by a politician did not hold. We can see from our experiment results for cross-training and testing on IBC and Convote that we got the worse performance, which is corroborated by the t-SNE visualization plots showing the separation between 2 datasets, rather than showing separation between 2 ideologies. This is due to the fact that sentences from Convote data were raw samples from US congressional floor debate transcripts and contained debate or conversational related words like "I yield", "Respected speaker" etc, whereas IBC contained proper human annotated ideology based sentences. An obvious next step after this project would be to implement recursive neural network to parse a sentence based on ideologies.

REFERENCES

[1] Somasundaran, Swapna, and Janyce Wiebe. "Recognizing stances in ideological on-line debates." Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text. Association for Computational Linguistics, 2010.

[2] Doumit, Sarjoun, and Ali Minai. "Online news media bias analysis using an LDA-NLP approach." International Conference on Complex Systems. 2011.

[3] Yu, Bei, Stefan Kaufmann, and Daniel Diermeier. "Exploring the characteristics of opinion expressions for political opinion classification." Proceedings of the 2008 international conference on Digital government research. Digital Government Society of North America, 2008.

[4] Preoiuc-Pietro, Daniel, et al. "Beyond binary labels: political ideology prediction of twitter users." Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2017.

[5] Misra, Arkajyoti, and Sanjib Basak. "Political bias analysis." (2016): 8.

[6] Socher, Richard, et al. "Parsing natural scenes and natural language with recursive neural networks." Proceedings of the 28th international conference on machine learning (ICML-11). 2011.

[7] Iyyer, Mohit, et al. "Political ideology detection using recursive neural networks." Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Vol. 1. 2014.

[8] Maaten, Laurens van der, and Geoffrey Hinton. "Visualizing data using t-SNE." Journal of machine learning research 9.Nov (2008): 2579-2605.

EEL-6935 Machine Learning for Natural Language Processing

April 27th, 2019

Political Ideology analysis with Natural Language Processing

Avanti Bhandarkar
Keerthiraj Nagaraj
Dept. of Electrical and Computer Engineering
avantibhandarkar@ufl.edu
k.nagaraj@ufl.edu

Project goal

- Our objective is to classify text sentences based on the political ideology that it falls under.

- We employ principles of Natural Language Processing and Machine Learning to address text sentence classification problem.

- We have conducted variety of experiments with 2 datasets, 3 features and 6 learning models to understand the correlation between political party affiliation and political ideology.

Dataset description

We have used two different dataset in our project

1. Ideological Book Corpus (IBC) dataset - labelled based on ideology
2. Convote dataset - labelled based on party affiliation

Statistic	IBC	Convote
Number of liberal sentences	2025	3711
Number of conservative sentences	1701	3708
Total number of words	85487	1016800
Total number of characters	628655	7330698
Average word length	5	5
Median words per sentence	22	42

TABLE I: Data statistics

Preprocessing

Preprocessing involves transforming raw text data to a form which is recognizable by natural language processing algorithm.

- Cleaning - removing unwanted part of text like punctuations and stopwords
- Normalization - Translating words with different POS to root form
- Lemmatization - converting words to their root form based on vocabulary and morphological analysis

Feature Extraction

We have worked with 3 different feature representations.

- Bag of Words - Number of times a particular word appears in document
- Term Frequency Inverse Document Frequency (TF-IDF) - gives less importance to words which have high frequency in all documents
- Bag of Words + TF-IDF: Combining the two feature representations

Model description

We have used 6 different models in our project

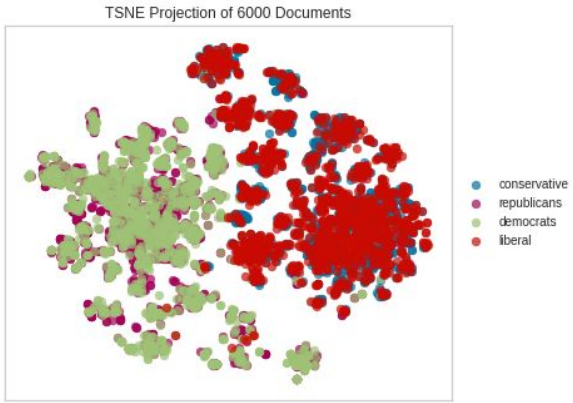
1. Machine Learning models - SVM, NBC, Logistic Regression
2. Deep Learning models - RNN, LSTM, GRU

Parameter name	Value
Recurrent layer	20
Dense layer	10,1
Activation function	Sigmoid
Loss function	Binary Cross entropy
Optimizer	ADAM
Validation patience	3 epochs
Dropout Ratio	0.3
Early Stopping	Yes
Number of epochs	10

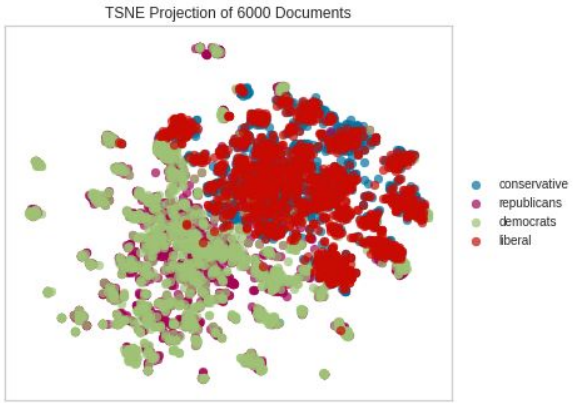
TABLE II: Deep Learning architecture

Experiments and results

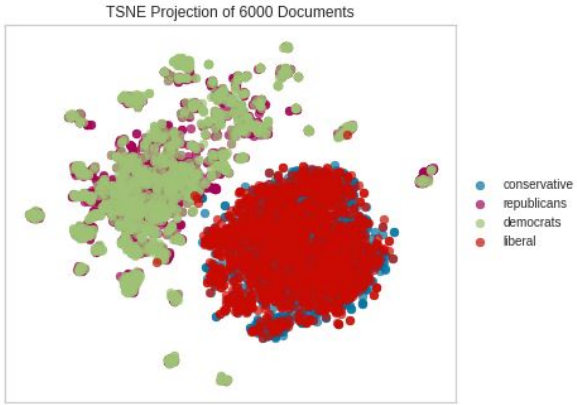
Experiment 1: t-SNE visualization in feature space



BOW for IBC+Convote data



TFIDF for IBC+Convote data



BOW+TFIDF for IBC+Convote data

Experiments and results

Experiment 2: Classification with IBC dataset

- Our objective was to develop models that can classify text sentences based on political ideology.
- For machine learning models, the highest test accuracy that we could obtain is 63.00\% with Naive Bayes Classifier and Bag of Words feature
- For deep learning models, the highest test accuracy is 60.19\% with Gated Recurrent Units.
- Low accuracy is due to small size of data, unbalanced class distribution and bias during labelling as ideology is subjective in nature.

Experiments and results

Experiment 3: Training with Convote data

- Our objective was to develop models that can classify text sentences based on political party affiliation.
- For machine learning models, the highest test accuracy that we could obtain is 72.37\% with Naive Bayes Classifier and TF- IDF
- For deep learning models, the highest test accuracy is 67.79\% with LSTMs, although GRU also results in similar performance with an accuracy of 67.25\% on test dataset.
- Higher performance is due to larger data size & balanced class size.

Experiments and results

Experiment 4: Cross testing

- In the literature, we noticed that Convote dataset was used for political ideology analysis but the dataset was labelled based on the party affiliation,
- It is based on the assumption that party affiliation and political ideology are very closely related.
- To test this assumption, we conducted 2 set of experiment:
  - Training on IBC and testing on Convote (TITC)
  - Training on Convote and testing on IBC (TCTI)
- Performance results doesn’t satisfactorily support this assumption

Experiments and results

Experiment 5: Combined dataset

- We created a dataset combining both IBC and Convote datasets
- We conducted this experiment to again the test the assumption that political ideology and party affiliation has high correlation as claimed in the literature.
- For machine learning models, the highest test accuracy that we could obtain is 66.44\% with Logistic Regression and BoW+TFIDF feature, and for deep learning models, the highest test accuracy is 63.62\% with LSTMs.

Experiments and results

Experiment 6: Semi Supervised Learning (SSL)

- We used labels from one dataset for the initial model training and pseudo labelling of samples of another dataset with this model.
- Trained a new neural network model with initial training set and pseudo labelled samples.
- Compared the test performance with and without SSL.
- With SSL, we were able to improve the test accuracy as shown in Tables 3 and 4.

Performance results

Following tables show test accuracy for all the experiments in our project

Experiment	SVM	NBC	LogReg
BOW IBC	62.60	63.00	59.78
BOW Convote	72.23	69.74	65.97
BOW Overall data	68.28	68.10	62.09
BOW TCTI	54.50	57.59	50.94
BOW TITC	52.20	50.99	51.57
TFIDF IBC	61.79	61.26	59.51
TFIDF Convote	72.37	71.56	71.36
TFIDF Overall data	67.56	68.50	66.44
TFIDF TCTI	55.28	56.98	52.60
TFIDF TITC	51.30	50.81	51.21
BOW +TFIDF IBC	62.19	60.00	57.64
BOW +TFIDF Convote	70.61	68.80	70.75
BOW +TFIDF Overall data	67.56	68.50	66.44
BOW +TFIDF TCTI	55.28	56.98	52.60
BOW +TFIDF TITC	51.30	50.81	51.21

TABLE III: Performance of Machine Learning models

Experiment	LSTM	GRU	RNN
IBC	57.77	60.19	56.17
SSL with IBC	65.01	61.13	57.64
Convote	67.79	67.25	63.48
SSL with convote	68.94	68.13	66.78
Overall data	63.62	62.49	55.85
TCTI	54.48	54.37	51.15
TITC	50.73	51.66	50.98

TABLE IV: Performance of deep learning models

Conclusions

- We have explored 2 widely used datasets for text sentence classification based on political ideology.
- We conducted extensive experimentation with different combination of feature representations and learning models
- We tested the commonly held assumption that political party affiliation and political ideology has high correlation.
- We were able to develop simpler models that could perform better than more complex models from the literature.

References

- Somasundaran, Swapna, and Janyce Wiebe. "Recognizing stances in ideological on-line debates." Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text. Association for Computational Linguistics, 2010.
- Doumit, Sarjoun, and Ali Minai. "Online news media bias analysis using an LDA-NLP approach." International Conference on Complex Systems. 2011.
- Yu, Bei, Stefan Kaufmann, and Daniel Diermeier. "Exploring the characteristics of opinion expressions for political opinion classification." Proceedings of the 2008 international conference on Digital government research. Digital Government Society of North America, 2008.
- Preoţiuc-Pietro, Daniel, et al. "Beyond binary labels: political ideology prediction of twitter users." Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2017.
- Misra, Arkajyoti, and Sanjib Basak. "Political bias analysis." (2016): 8.
- Socher, Richard, et al. "Parsing natural scenes and natural language with recursive neural networks." Proceedings of the 28th international conference on machine learning (ICML-11). 2011.
- Iyyer, Mohit, et al. "Political ideology detection using recursive neural networks." Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Vol. 1. 2014.
- Maaten, Laurens van der, and Geoffrey Hinton. "Visualizing data using t-SNE." Journal of machine learning research 9.Nov (2008): 2579-2605.