Political Ideology Detection by RNN-LDA Model

Haonan Tian
Master Student
Center for Data Science
New York University
New York, NY 10011
Email: ht1151@nyu.edu

Abstract With the raise of data science methods, the analysis of political ideology for any speaker or document has never been restricted to literal analysis. Some probabilistic model such as LDA has already enabled people to automatically generate abstract structures from large size corpuses. However, the traditional probabilistic methods only focuses on the probabilistic measures of corpus and seldom take the underlying contextual meanings of corpuses into consideration which makes them less competitive in multiple text analysis scenarios such as text classification based on contextual meanings.This paper proposes a new method which combines the LDA topic modeling methods with the recurrent neural network. The target of this model is to accurately classify the speeches in the Congressional Speech Corpus by the parties to which the speakers of the speeches belong The results of the model is compared with a baseline model which applies Naive Bayes classification methods. Word Count: 2838

1 Introduction

This paper explores the possibilities of applying data science methods to determine the ideology position of a given document. As the volume of the data capturing human communication has risen to match the goal of applying computational text analysis to understanding human behaviors and societies, the automated computational models to extract critical insights from all these unstructured data has been throughly implemented in social science [1].

With the rapid growth of machine learning methods in natural language processing, the text analysis methods have been applied to multiple research domains such as biology, social science etc. Some powerful methods such as topic modeling and k-means clustering have made it convenient for people to mine the contextual structure of a corpus from statistical perspectives. Additional text analysis algorithms have made it possible to make classification analysis based on large volumes of text data. The recent development of natural language processing techniques nevertheless, has dramatically change the method people commonly applied to conduct text analysis. The increase of the computational

power for the modern distributed computational system on the other hand, have paved the way for the application of these complicated algorithms. A popular topic in natural language processing is sentimental analysis which has been widely applied for industries. However among all those classification algorithms, few have been applied to make analysis on the political ideology classification. Although the R package WordFish has provided a practical method for identifying the documents? political positions, more research is required to explore the potential models people may use to detect the political ideological position for certain given documents. Therefore, in this paper, a new model will be proposed to perform classification analysis on document with nearly half of labeled as democrat and the other half labeled as republican. The model will combine the topic modeling methods with deep learning methods to make predictions. Specifically, a LDA model will be trained and the trained model will make topic distribution prediction for each document within the corpus. The output vector will then serves as the partial input for the deep learning model to make prediction. Since one apparent disadvantage of LDA model is that it ignores the contextual meaning of the documents themselves and generated the abstract topics from the statistical distribution of words among documents, while deep learning models such as Recurrent Neural Network are powerful at detecting the contextual meaning of given texts, it is believed that the combination of two will generate better performance for classification on ideology positions.

2 Literature

It is admitted that the automated computational models have been applied to large numbers of research areas. Political science relies on the study of a large volume of documents and is the ideal domain of the application of these computational models. Currently, the most common application of the computational models used in political science is for the election prediction. Other common applications include government agenda detection, policy prediction, etc. Few research has been conduct to measure the ideology attitudes contained in people?s words.

As for the methodology, most research involve text analysis in political science focuses on applying simple methods such as word counts, word frequencies, documents TTR (type token ratios) to extract the features from documents. More computationally complex models such as k-means clustering, LDA (Latent Dirichlet allocation) tries to extract the abstract structure from the documents. However, all of these methods ignores the underlying contextual meaning of the documents and are very sensitive of preprocessing steps. According to Iyyer et al, The current text analysis approaches on ideology detection have not gone far beyond bag of words classifier thus neglecting the richer linguistic context of this kind and often operated at the document level [9]. The other research conducted by Misra and Basak applied deep learning methods to extract political bias from textual dataset. All these research have demonstrated that deep learning methods are good at extracting contextual meanings from text data and will outperform traditional text analysis methods.

In this paper, an ensemble method is proposed to make political ideology prediction based on text data. The data used is the Congressional Speech Data. The data contains the debates and arguments made by congressmen with the corresponding bills and the speakers identities. The model is trained to make prediction to tell if the arguments are made by speakers from democrat or republican.

3 Theory and Hypothesis

The standard LDA model is an unsupervised learning method which considers each document in the corpus as a bag of words. It randomly assigns the weights to topic for

each document and applied reverse engineering to make improvements on the distribution of the topics across the documents. The improvement is made by reassigning the probabilities of the topic assignment on each words which depends on the topic distribution of the document as well as the topic distribution of words across all documents. Therefore, the LDA model is essentially a probabilistic model which only takes the term frequencies and cross-document frequencies of words into consideration. Although the topic distribution returned by LDA model is concise and easy to interpret, the model?s performance is hard to measures. Besides, the model fails to catch the underlying contextual meanings of documents which makes it less competitive of extracting contextual meanings from corpus.

The deep learning model however, is robust at catching contextual meanings of text data. Although most of the deep learning methods suffer from poor interpretability, they have became the most popular methods for natural language processing. A typical deep learning method is the recurrent neural network (RNN) which is effective at processing sequential data. The RNN model applies recursive computation unit regarding each input conditioned on the inputs of previous states. The latent embedding layer in the RNN model can be considered as the summary of all the inputs from previous steps. The RNN has been widely used in typical natural language processing problems such as machine translation, language modeling etc.

The model proposed in this paper combines the LDA model with the deep learning model. Since LDA models are powerful at summarizing the underlying structure of the corpus whiling RNN models are strong at catching contextual meaning of input tokens, I believe the combination will inherits the strengths from both models so that it will better represents the contextual meanings of document at word-level and at the same time, catch the features summarized as topic distribution at document level. Since the target of the model is to make ideology classification for each debate argument recorded as text blocks, it is believed that the model proposed can better classify the data compared with traditional classification model such as Naive Bayes.

4 Data and Methods

The data used in this model is the Congressional Speech Data (Convote)[3]. The data records the congressional floor debate transcript from 2005 in which all speakers have been labeled with their political parties [2]. The data is downloaded from US house record and has been preprocessed so that documents is labeled by the bills number, speakers? id and their corresponding parties. Since the data is originally collected for congress bill vote result prediction, the whole data has been divided into three stages with data for each stage differs in the document naming protocol. However, the slight difference will not influence the analysis conducted in this paper and therefore, and only the stage one data has been selected as the data to use in this paper since the majority of stage two and stage three documents are covered by stage one documents. The main reasons for selecting this data are that the similar research which involves applying RNN model to prediction binary ideology from text documents has used this dataset. Additionally, the debate document is the typical text files which directly record the speaker?s expressions. It is proper to assume that the speaker on the debates will argue for the ideologies supported by the parties they belong to. Therefore the congressional records are the ideal corpus for ideology analysis. Finally, since few research has been conducted on political analysis with data science methods, the Convote labeled data can be served as an ideal resource for supervised learning.

The following sections describe the detail implementations of the establishment of the model.

4.1 Data Preprocessing

Since LDA model is sensitive to preprocessing methods applied on the input corpus, the preprocessing steps are carefully chosen in the analysis. The data is loaded into the program and the party name (democrat or republican) associated with each document is recorded as the label for the document. Each document is tokenized at word level. The punctuations and numbers have been removed. Then all tokens is lowercased, and stop words are removed. Finally, the tokens are stemmed.

4.2 LDA Model

The LDA model is performed by using the python gensim package. The preprocessed training set data is used to build the token vocabulary then according to the index assigned to each token, every document in the corpus is converted into a numeric vector with the tokens replaced by corresponding indices. The the converted corpus is used to train the LDA model provided by gensim package. Then the trained model can take any new document and return the topic distribution of the input document. The trained model is saved for later use in deep learning model.

4.3 Naive Bayes Model

A Naive Bayes model is established in the program which serves as the baseline model. The strong assumption of Naive Bayes that all tokens in the document are independent makes it a computationally simple algorithm. However, it has been demonstrated in large numbers of practical cases that Naive Bayes always provides decent accuracies with shorter training time. The Naive Bayes model used here has two versions. The first version is the count-based model while the second one is the tfidf-based model. The main difference of these two models is that in the first version of model, tokens in the corpus are represented by their frequencies in the document while the tfidf-based model takes the not only the term frequencies but also the inverse document frequencies into consideration. In other words, if a certain token only contained in a certain document, then although this token may have low term frequency, it will be assigned a large weight regarding the inverse document frequency.

4.4 Proposed RNN-LDA Model

This is the model proposed by this paper. The model is comprised of a pre-trained LDA model, an RNN model and a fully connected multi-layer perceptron model. Specifically, the LDA model is trained by the training data so that it can return the topic distribution of any given input. Then the RNN model is trained by training data. The return of the RNN model is an embedding vector which is the representation of the input sequential data. The returned embedding layer will then be concatenated with the topic vector return by LDA model. The combined vector will then be input to the multi-layer perceptron model which will make the prediction.

5 Results

5.1 Data Exploration

For the training set, there are 5634 speeches and 2786 of them are labeled as conservative and 2848 of them are labeled as liberal. For the testing set, there are 702 speeches within which 367 are labeled as conservative and 335 are labeled as liberal. For the training data, the average length of the speeches before text preprocessing is 407 words while the average length of speeches after preprocessing steps is 137 tokens. The following graphs displays the distribution of lengths and total number of unique tokens in the speech corpus.

The Type-Token Ratio is calculated for each speech and the distribution is shown below:

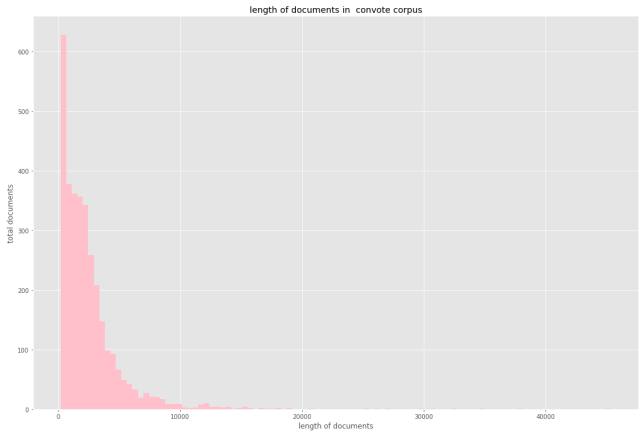

Fig. 1. Speech Length Distribution

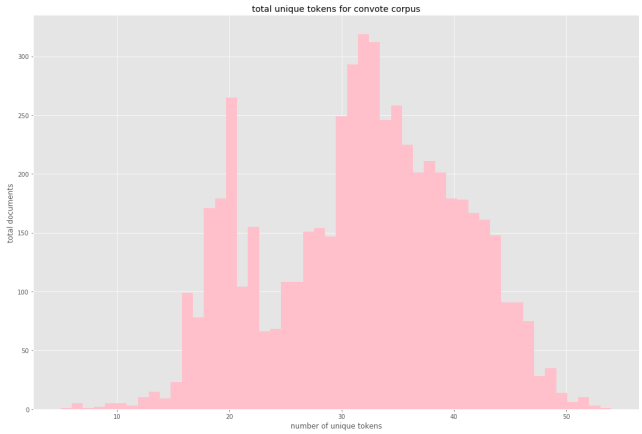

Fig. 2. Distribution of Total Number of Unique Tokens

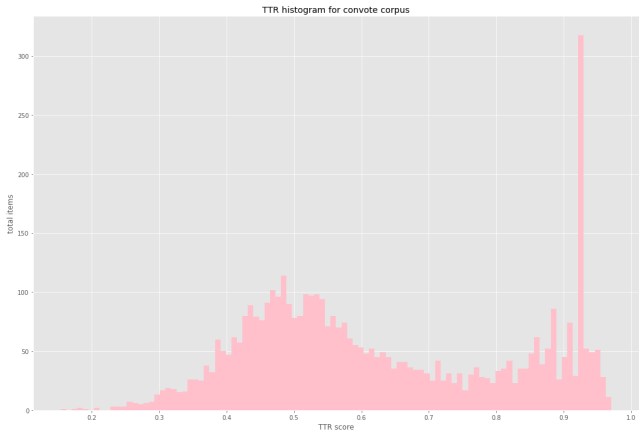

Fig. 3. Distribution of TTR

As can be seen from the above graphs, there are two peaks of TTR in the corpus and with a peak around 0.5 and the other one around 0.85. This may indicate that there are two distributions of speeches measure by vocabulary variations.

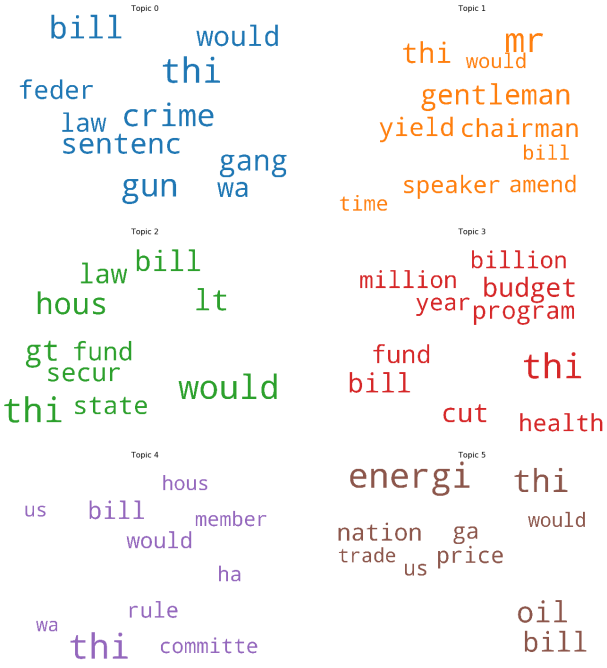

Fig. 4. Word Cloud For LDA Model on Liberal Set

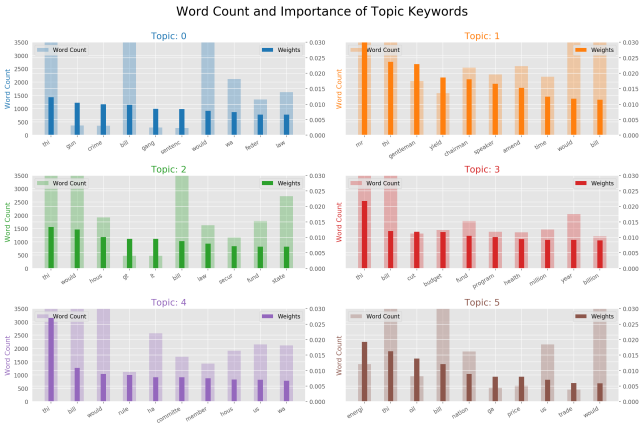

Fig. 5. Word Count For Key Words on Liberal Set

5.2 LDA Model

For this section, the training set is divided into two sets according to the labels of the speeches (either conservative or liberal). The LDA model is performed on conservative subset, liberal subset and the whole training set. The number of topics to generate are set at 6. For each set, a wordcloud gragh, a word count graph for key words and a t-SNE clustering chart are generated to better visualize the features for LDA model based on each set.

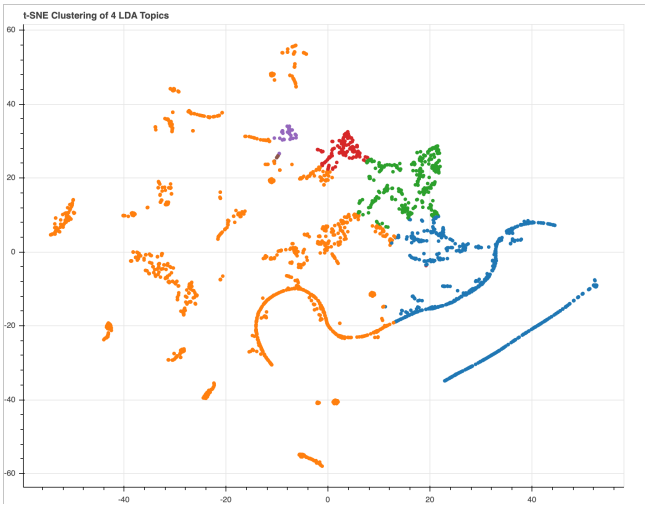

Fig. 6. t-SNE For LDA Model on Liberal Set

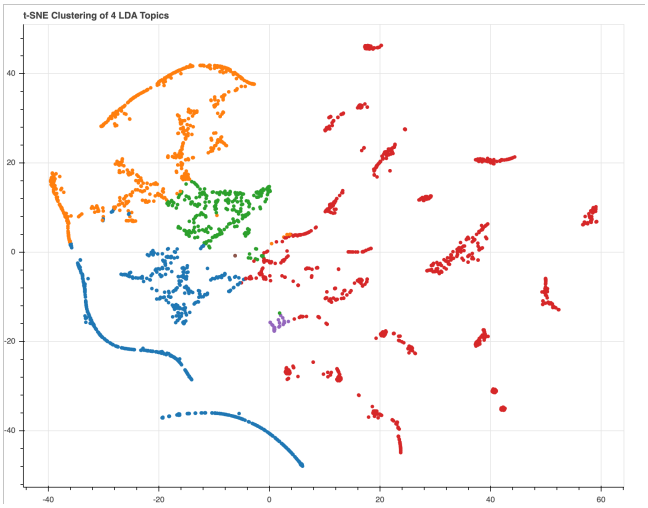

Fig. 9. t-SNE For LDA Model on Conservative Set

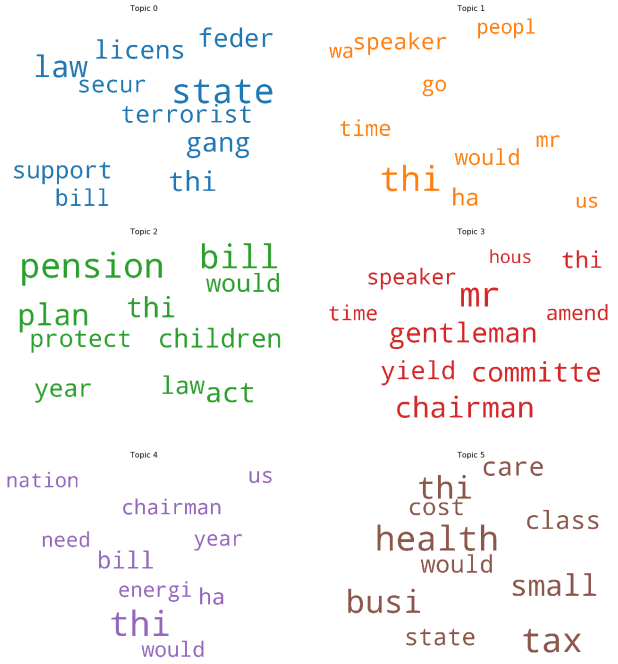

Fig. 7. Word Cloud For LDA Model on Conservative Set

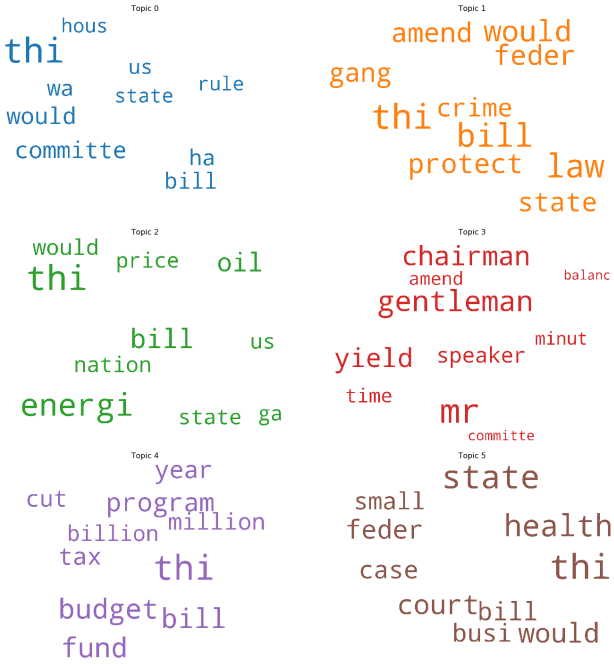

Fig. 10. Word Cloud For LDA Model on Training Set

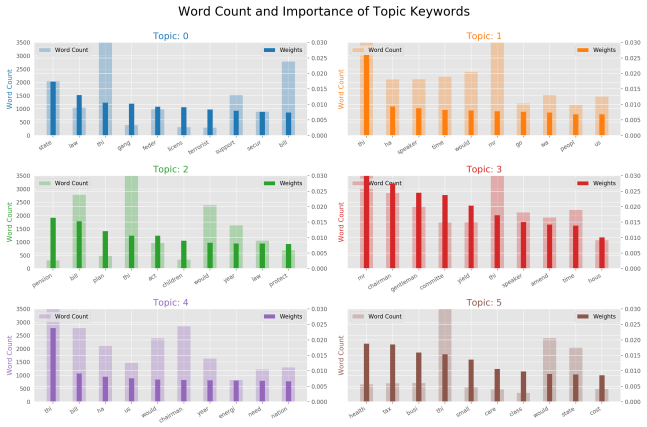

Fig. 8. Word Count For Key Words on Conservative Set

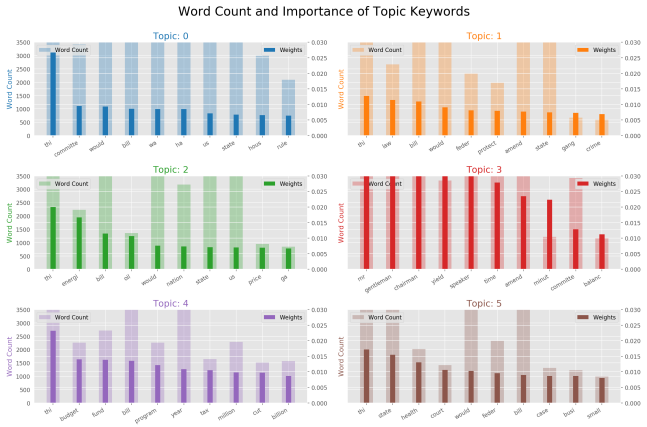

Fig. 11. Word Count For Key Words on Training Set

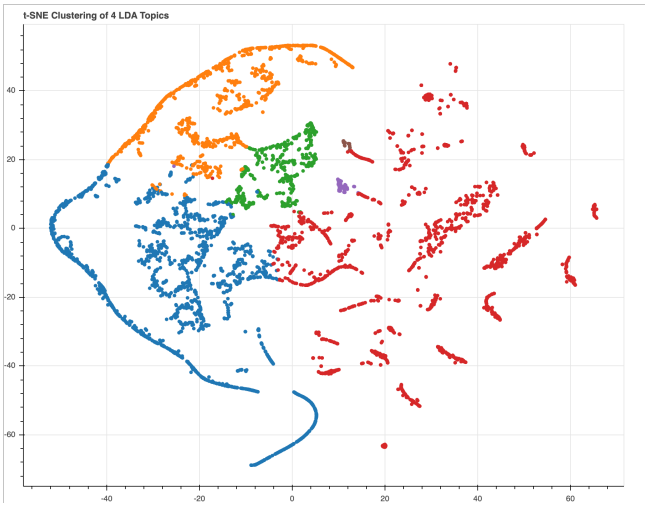

Fig. 12. t-SNE For LDA Model on Training Set

5.3 Naive Bayes Model

Text preprocessing is essential for any text analysis algorithms. The Naive Bayes model trained in this program is based on two different preprocessing methods. One of these preprocessing steps is the count-based vectorization which converts the speech token list to a numeric vector with each entry represent the frequencies of the term. The other method applied is the Tf-idf method which also takes the inverse document frequency of terms in a vector into consideration. The accuracy for the count-based Naive Bayes model is 0.584 and the tf-idf-based Naive Bayes model reaches an accuracy of 0.629. The followings are two confusion matrices returned by these two algorithms:

Table 1. Count-based Naive Bayes Model Confusion Matrix

	True	**False**
Positive	288	79
Negative	213	122

Table 2. Tfidf-based Naive Bayes Model Confusion Matrix

	True	**False**
Positive	246	121
Negative	139	196

5.4 RNN-LDA Model

Finally, various sets of parameters are tried on the RNN-LDA model (RL model) and the best parameter set is encountered when the vocabulary size is set at 20000 with the maximum sentence length set to 200. The Dimension of the embedding layer in RNN is set to be 200 and the dimension of the hidden layer of the fully connected neural network is set to be 50. The final accuracy of the RL model reaches 0.70.

6 Discussion

According to the results revealed in the previous section, the data sets used in this project are equally divided into two classes. From the LDA model, we can till that the top topics for liberal and conservative datasets are very different. The underlying structure revealed by t-SNE graph of the whole training set seems to be more similar to the conservative data set.

The Naive Bayes models established have similar accuracies. The tf-idf-based Naive Bayes model performs slightly better than the count-based model. The probable explanation is that the inverse term frequency matters. Although all texts in the corpus have been preprocessed when sent to the Naive Bayes model, the sparse tokens for any speech still contains more information than the tokens which are common across all speeches. Although Naive Bayes model is computationally efficient and practically effective, the results are slightly better than random guess. This indicates that the model has high bias for this problem.

An apparent drawback of the RNN-LDA model is that the selection of the hyper-parameters such as the hidden dimensions of RNN or fully connected neural networks are chosen empirically and the hyper-parameter selection is extremely difficult since the training requires large computational power. As for the data fed into the model, the vocab-

ulary size is chosen with a set of most frequent tokens so that only these tokens will be converted to numerical values while the others are set with value corresponds to label unknown. The larger the vocabulary size, the more tokens will be included. In addition to the above hyper-parameters, the decisions are required to set the input dimensions of the text data, since we need to feed a fixed dimension to the RNN model. The longest speech in the corpus has length larger than 1000, and it is insensible to maintain this length and pad every other speech to match with this length. The length of the speech we set as input to the RNN model is actually the time steps each RNN computational unit will walk through to update the embedding layers. The longer sentence length we set, the more information each RNN unit needs to catch.

The results of the RL model is better than the traditional Naive Bayes model. However, the result is far from decent. The probable reasons for the low accuracy may be: 1. the hyper-parameter set is still not the most effective one. Due to the limitation of the computational power, only 4 sets of hyper-parameters are tested with 200 epochs. It is suggested that a grid search may be performed to mine the optimal set of hyper-parameters. 2. Since the RL model is a very complicated model, the low accuracy may due to the lack of data. Since we only have around 5000 speeches and some of them are very short after preprocessing, there may be no enough data to train the model. 3. there may exit flaws in the design of the model. The RNN model is based on basic LSTM units and the output of RNN model is simply concatenated with the output topic vector from LDA model, and the whole representative vector is fed into the fully connected neural network with one hidden layer. Maybe there is smarter way to combine this two vectors. It will be possible to insert additional hidden layers to the model.

7 References

[1] O Connor, Brendan & Bamman, David & A Smith, Noah. (2012). Computational Text Analysis for Social Science: Model Assumptions and Complexity. 41.

[2] Iyyer, Mohit et al. ?Political Ideology Detection Using Recursive Neural Networks.? ACL (2014).

[3] Thomas, M., Pang, B., & Lee, L. (2006). Get out the vote: determining support or opposition from Congressional floor-debate transcripts. In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, (pp. 327-335). Sydney:ACL. Retrieved from http://www.aclweb.org/anthology/W/W06/W06-1639, http://www.cs.cornell.edu/home/llee/papers/tpl-convote.dec06.pdf