## Political Text Topic Extraction - NLP Group 3 Final Project

**Team:** Daniel Liu, Aaryan Virani, Brian Zou, Maddy Li, Laura Chen

# Problem History/Introduction

A major recurring theme during the past 2024 election season is “information”. More specifically, misinformation or “too much” information - the point where voters don’t know how to filter different sources and gain a better understanding of the overall picture. Past research has found that most voters vote based along the lines of their political ideologies. Additionally, most people do not know their election candidates well enough, specifically on the local level.

As a result, we propose a research topic focused on topic modeling/categorizations for political texts. The objective is to develop a system that can extract meaningful information from political texts, and categorize them based on their relevance to predefined categories. A model that can filter political texts and categorize them correspondingly can be of large significance, especially for voters who are overwhelmed by the large influx of data present in the 21st century.

# Strategy for Solving the Problem

Currently, we have several ideas on what information we want to extract from a political text. We proposed several approaches, and we plan to narrow them down as we begin to create our models and move along in our research. One potential approach is to extract the political issue that a document discusses, such as “gun control”, “healthcare”, “abortion”, etc. This can be extended to extract not just the issue, but also the politician’s stance on the issue - this idea is echoed in several papers listed in the literature review section, where the papers proposed models to extract political topics and stances.

Another categorization is to determine a political document as “high risk”, “neutral”, and “low risk”. Especially during the 2024 election season, we have seen a much larger influx of inflammatory speeches, news headlines meant to grab attention rather than information, and more. By leveraging NLP techniques like TF-IDF (analyzing frequencies of inflammatory words/messages), BERT embeddings and topic modelings, the model can provide insights into a political document’s “riskiness”, which can be very beneficial to today’s inflammatory political stage.

Our research methodology will compare several different topic modeling approaches, and evaluate which model is most suitable for topic modeling political texts. Below, we briefly describe several approaches we want to consider in our research.

## Data Collection, Annotation and Preprocessing

**Data Collection**: We plan to focus our research on the 2024 election news articles and political speeches. On political speeches, we can gather them from a from online sources or transcribe them from video recordings. On political news articles, we can obtain them in large quantities through NYU’s database, specifically ProquestTDM. Additionally, we will likely use web scraping tools or APIs to automate data collection from sources such as Guardians and the New York Times.

**Annotation**: To create our training dataset, we need to annotate the news articles per our defined categories. Specifically, for each news article, we will label its political issue and stance. A political issue could be wide-ranged, but a stance will be on a one-dimensional scale (pro or against), for the political issue. (Note: past research papers suggest that political stance is more complicated than sentiment analysis, or a "pro" and "against" in general. As a result, we will specify it in the context of the political issue that the news article discusses). Additionally, we will label the news article as "high risk" and "low risk" to explore potential categorizations in that area.

**Preprocessing**: Once collected, we will preprocess the text data to remove irrelevant information such as HTML tags, non-informative words (e.g., "thank you," "great service"), and non-text elements. We will standardize the text through tokenization, lowercasing, and stopword removal to ensure consistency across entries.

Additionally, we want to also consider "chunking" the data. Documents like news articles and tweets usually discuss one issue per document. However, political speeches may touch on several topics per speech. We may want to separate speeches into chunks where each chunk discusses a specific topic.

## Model To Explore

Below are the models we want to explore in our political text topic extraction research project. We will begin by analyzing the base models using NMF, and build on these models through extensions such as models proposed in the research papers (listed in the literature review section below).

### NMF (Base Model)

Nonnegative Matrix Factorization as an unsupervised machine learning algorithm to determine the topics and the word topic distribution. We use the trained topic word distribution to predict a new document's topic. Additionally, we can also consider LDA as a base model

NMF is a technique that decomposes a large matrix into lower-dimensional matrices, which can give us the "topics" within the data. I chose to use a linear algebraic approach instead of a probabilistic model like Latent Dirichlet Allocation (LDA) because it is computationally faster and relatively straightforward to implement.

To implement NMF, we will do the following steps. First, the cleaned, preprocessed text will be converted into a document-term matrix (DTM), where rows represent individual customer reviews, and columns represent the unique words (terms). Using the DTM as input, we'll apply NMF to decompose the matrix into two matrices: a document-topic matrix which indicates the weight of each topic within each document and a topic-word matrix which indicates the weight of each word in each topic. We'll then experiment with different values to find the optimal number of topics (we can use human evaluation for this). Once our NMF is done, we can examine the top words within each topic to assign meaningful labels. These labels can be used to interpret themes in customer feedback, such as product quality, customer satisfaction, or value for money.

Post NMF Topic Modeling, we have two matrices: W and H, where W is the document-topic matrix and H is the topic word matrix (or the other way around, depending on if we start with a “document-term” matrix or a transposed “term-document” matrix). The document topic matrix creates a matrix such that each row represents a document, and each column represents a topic. Since the value represents the weight of each topic within each document, we can easily categorize the document to the topic with the highest weight.

The problem is, how can we classify new documents after we have trained a W and H for a given set of documents? For example, suppose we have a new document that can be represented by a document-term matrix of 1xn (1 document, n terms). During training, we have created a topic word matrix, where each value represents the weight of each topic for a word. We use the document-term matrix and the topic word matrix to estimate a new document-topic matrix *for this new document*. We follow the same method as above to determine the topic for this document.

Extension: Using KNN on the document topic matrix to classify new documents. The document topic matrix breaks down the *topic distribution* for a given document for some *t* topics (the number of columns in the DTM, which we define when we train the model). Let K equal this value (in other words, define t labels, and match each label to a “topic”). Then, we can train a KNN classifier on the document topic matrix, which will help *predict a new document’s label* for a document’s topic distribution. Afterward, once we have transformed a new document into the same TF-IDF (assuming we use TF-IDF) space and created an approximate document-topic matrix (or, more precisely, a document-topic vector since it’s just one document), we can use KNN to predict this document’s label. A benefit of this strategy is that this can use contextual information to predict a document’s topic based on its topic distribution, rather than solely looking at which topic has the higher weight.

## BERT Embedding and KNN

Use pre-trained models, such as BERT, to generate embeddings that can capture semantic information of texts. Then, we use clustering algorithms to extract relevant topics, and use KNN to predict new political text’s category.

Once the dataset is processed, each piece of feedback needs to be converted into vector format as has been aforementioned, such as BERT or other similar transformer-based models (RoBERTa, DistilBERT, GPT?). We can achieve this goal by generating embeddings for each feedback entry, which are dense vector representations capturing contextual information.

BERT captures nuanced meaning by processing each word within its context. It can understand both the direct meaning of words and their relation to surrounding words, thereby making it highly effective for capturing sentiment and focus of each feedback entry. It also seems to be a viable choice for it has generally outperformed other NLP models regarding recall.

Using a pre-trained BERT model, we can generate embeddings as follows:

- **Tokenization**: Each feedback entry is tokenized to convert words into tokens, which BERT understands. Tokenization includes adding special tokens [CLS] (Classification) to represent the beginning of an input sentence and [SEP] (Seperation) to mark the separation between different segments of text.

- **Encoding**: The tokenized text is then passed through BERT's layers. BERT generates a 768-dimensional vector (for the base version) for each token, providing rich contextual information.
- **Pooling**: To obtain a single embedding for each feedback entry, we would typically use [CLS] token's embedding, which represents the sentence's overall meaning. An alternative approach though could be that we average the embeddings of all tokens in the sentence for a slightly different representation.

**Output**: The output of the embedding process would be a dense, fixed-length vector for each feedback entry preserving contextual nuances of customer sentiment and topics. These vectors would serve as the basis for subsequent topic modeling and relevance scoring. The embeddings will allow us to capture clusters of similar feedback, and through relevance scoring, help us categorize feedback accurately

### Seeded-LDA

Rather than an unsupervised LDA approach where the ML algorithm determines the topics by itself, we take a Seeded LDA approach where we predetermine the categories, and the ML algorithm creates the topic word distributions based on the predefined categories.

### LDA Extension: Political Issue Extraction Model

The model proposed by Joshi, et al. The model builds on top of Latent Dirichlet Allocation and uses a hierarchical latent structure, where the variables are *issue* and *position*. Additionally, it assumes that each statement is made up of *issue words*, *position words*, and *emoticons*, and builds a hierarchical distribution/generative process based on this idea. Finally, the models distributions between each type of word and the latent variables (i.e. issue word-issue distribution, position word-issue-position distribution, etc.). Most importantly, word labeling (labeling words as "issue", "position", and "emoticons") and creating word distributions extends what we have learned in class.

We want to build on top of this method by combining it with other research models we have researched (see literature review), as well as specific information specific to the 2024 election season. This will help provide insight into whether model customization could play a factor in determining a model's accuracy.

## Visualization and Reporting

After all the previous parts are completed, we are ready to create an intuitive interface to summarize the results of our topic modeling and relevance scoring. This allows for the data to be consumed by end-users in an easy digestible format through providing quick insights and visualizations.

**Interface design**: The main part of the interface will be a dashboard that displays the key topics identified across political texts. This will be organized by displaying the main topics within political categorizations in an organized manner, with the most relevant on top so they are easy to identify. With each label, the interface will allow the user to select the topic and see additional information regarding it.

**Topic Trend Visualization**: In addition, we can add elements of data visualization to each topic to allow for easy identification of trends in relevance. For instance, line graphs or bar charts can show the frequency and relevance score of each topic across specific timeframes (weekly, monthly, quarterly). This will allow for a tailored view of the insights gathered through our program.

# Evaluation Plan

We will validate the topic extracted by the model against our standard topic list. To evaluate our system, we will perform the following steps:
1. *Task*: The system will categorize political texts into predefined topics as aforementioned and accordingly score their relevance
2. *Output Evaluation*: The performance will be measured using metrics that have been covered in class, such as accuracy, precision, recall, and F1-score for topic classification. MAP scoring to assess the effectiveness of relevance scoring.
3. *Data Partitioning*: The dataset as we see it, will be partitioned into the following, with 70% going into training, 15% going into development, and 15% going into test sets (however, this is definitely subject to change). We will use cross-validation during training but keep the test set separate to ensure unbiased evaluation.
4. *Error Analysis*: Error analysis will involve a detailed examination of the model's predictions on the development set to identify patterns in errors and areas for improvement. Misclassified texts will be categorized to determine if errors arise from ambiguous language, overlapping topics, or insufficient training data for certain categories. Additionally, we will analyze errors by topic to detect potential biases in the model's predictions or dataset imbalances. Insights gained from this analysis will guide feature engineering, model adjustments, ultimately enhancing overall system performance.
5. *Validity Assurance*: To ensure our results are valid we will use techniques like cross-validation. Additionally, in the case of manual labeling tasks (which does not seem to be necessarily applicable at least in this stage), we will measure inter-annotator agreement, however, this is as well subject to change.

# Literature Review

The following academic articles have been reviewed to inform our project approach. These are potential articles we have considered, however, again, they may not be used as reference on the final draft:

1. *Political Issue Extraction Model: A Novel Hierarchical Topic Model That Uses Tweets By Political And Non-Political Authors (Joshi et al., WASSA 2016)*. This paper presents a model to extract information from tweets, specifically the political issue that the tweet is about, and the author's position. This paper provides a good model that we will reference in building our model to extract political issues/topics from political data sources. Additionally, it provides insight on how to evaluate such models. *(*https://aclanthology.org/W16-0415/ *)*
2. *Mining contrastive opinions on political texts using cross-perspective topic model (Fang, et al., ACM 2012)*. Early paper that proposes an unsupervised topic model for contrastive opinion modeling. This paper proposes to separate the process for topic generation and word generation, thereby creating topic word distribution and a separate *opinion word distribution* for each topic

(i.e. “healthcare based on private insurance” vs “healthcare based on public, government-funded health agencies). (*https://www.cse.scu.edu/~yfang/wsdm12-com.pdf*)

3. *A Bayesian Hierarchical Topic Model for Political Texts: Measuring Expressed Agendas in Senate Press Releases (Grimmer 2009, Harvard University)*. This paper introduces a Bayesian Hierarchical Topic Model to analyze political texts, with a focus on U.S. Senate press releases. It looks at how legislators discuss their priorities, and discusses clustering documents based on topics. (https://web.stanford.edu/~jgrimmer/ExpAgendaFinal.pdf)
4. *Interactive Topic Modeling (Hu et al., ACL 2011)* - The piece introduces a semi-supervised approach to topic modeling, highlighting human-in-the-loop interaction, which could align with our needs of requiring certain customization/human touch to fine-tune the results (looking at **LDA** or **NMF**) (https://aclanthology.org/P11-1026/)
5. *TopicGPT: A Prompt-based Topic Modeling Framework (Pham et al., NAACL 2024)*- This article leverages the use of LLMs such as ChatGPT to enhance topic extraction. Though the article as whole may not be so relevant, one key element that can be pulled is that the authors demonstrate how prompts can be used to guide topic modeling, which could help us refine the topic extraction. (https://aclanthology.org/2024.naacl-long.164/)
6. *A User-Centered, Interactive, Human-in-the-Loop Topic Modelling System (Fang et al., EACL 2023) -* This article presents a user-centered system that allows for interactive topic modeling, emphasizing **visualization**. It can allow us to make the topic analysis results more interpretable for end users. (https://aclanthology.org/2023.eacl-main.37/)

# Group Work Division:

The work will be approximately divided into the following sections.
**Brian Zou**: Data Collection and Preprocessing - sourcing, cleaning, and normalizing the political text dataset.
**Aaryan Virani**: Embedding and Vectorization - implementing BERT or similar embeddings to represent feedback, and political text categorization using KNN following the BERT embeddings.
**Laura Chen**: Topic Modeling - conducting topic analysis and identifying topics in political text using methods like LDA or NMF.
**Daniel Liu**: Topic Modeling/Relevance scoring - conduct topic analysis (different than the base model) and identify topics in the political text. Determine models/metrics to calculate the relevance score between a new document and a political category
**Maddy Li**: Visualization and Reporting - creating an interface to display results and summaries for user insights.