

Sentiment Analysis Models – Researching Rule-Based, Machine Learning, and Transformer Models

AP Research & Senior Research Project

Ojas Chaturvedi

<https://github.com/ojas-chaturvedi>

This week marked the completion of my classification script execution on all legislative text data from 2001 to 2023. Each piece of legislation has now been classified as either ‘control’ or ‘rights’. Due to the OpenAI rate limits, which I previously discussed in [last week’s blog](#), I had to operate the script continuously on my laptop throughout the week. Nevertheless, I successfully processed all the relevant legislation within the set timeframe: 2001 - 2023.

Legislative data (with classification update):

https://github.com/ojas-chaturvedi/NLP-Gun-Legislation/tree/master/data_collection/data

Simultaneously, I initiated the development of my sentiment analysis models. These models are designed to analyze the text of the legislation and output a sentiment score ranging from -1 to 1 . Scores of 0.05 or higher denote positive sentiments, scores of -0.05 or lower indicate negative sentiments, and scores in between suggest a neutral stance. Prior to modeling, I conducted a thorough review of existing sentiment analysis technologies to understand the advantages and drawbacks of various approaches. From this research, I have identified three main types of sentiment analysis models that are most suitable for my project.

The first approach I explored is the Rule-based model, a traditional method that relies on pre-defined rules and lexicons to determine sentiment. In this model, polarity scores are assigned to specific words or phrases; for instance, positive scores might be attributed to words like “love”, “happy”, and “amazing”, whereas negative scores would be given to “hate”, “sad”, and “terrible”. The model then aggregates these scores to derive the overall sentiment of the text. Although this method is straightforward, its major drawback is its inability to detect nuances such as sarcasm—a limitation generally not critical in the context of legislative texts, which are expected to be straightforward and unambiguous.

The second type of model is the Machine Learning model, which operates on principles of pattern recognition and feature-based predictions. These models can be divided into supervised and unsupervised categories. Supervised models, such as logistic regression, support vector machines, or neural networks, are trained using labeled data to classify text into predefined sentiment categories like positive, negative, or neutral. On the other hand, unsupervised mod-

els—such as clustering, topic modeling, or word embeddings—identify underlying structures and meanings in texts without relying on pre-labeled data. Machine learning models offer greater flexibility and power compared to rule-based models, adapting more dynamically to the complexities of language used in legislative texts. However, they also come with their own set of challenges. They require significant amounts of data and computational resources, are susceptible to biases introduced by flawed data or poorly chosen features, and their internal decision-making processes can be opaque and difficult to interpret. This complexity can make them less accessible for straightforward applications but highly effective for deep, nuanced text analysis.

The last major type of sentiment analysis model is the Transformer model, which represents the cutting edge in deep neural network technology. Transformers use an “attention mechanism” that helps the model understand the relationships and dependencies between words and sentences across large texts. This capability allows Transformers to process and analyze text in parallel, providing a robust understanding of context, semantics, and nuanced linguistic elements. Transformer models can be categorized as either pre-trained or fine-tuned. Pre-trained models, such as those built on datasets like Wikipedia or books, learn a general representation of language that can be applied broadly. Fine-tuned models, however, are adapted from these general frameworks to perform specific tasks or understand particular domains—like analyzing tweets, movie reviews, or legislative texts—using a more focused dataset. While Transformer models are among the most effective tools for sentiment analysis, they come with significant challenges. They require extensive data and computational power to train effectively. Additionally, due to their complex nature, they can produce errors or inconsistencies, particularly if the training data has limitations or the model setup is not meticulously tuned. Moreover, the “black box” nature of these models often makes them difficult to interpret or trust, posing challenges to the validation and explanation of their decision-making processes. The “black box” nature refers to situations where the internal workings of the model are not transparent or easily understandable, even to those who design and operate the model, complicating efforts to verify the reasons behind its predictions and assess its reliability.

This week, I began working with rule-based sentiment analysis models, starting with VADER (Valence Aware Dictionary and sEntiment Reasoner). VADER enhances the traditional sentiment lexicon approach by providing a larger, yet straightforward and swiftly deployable tool that does not require extensive training. It is readily inspected, easily understood, and can be seamlessly extended. Furthermore, VADER accounts for word-order sensitive relationships that significantly affect sentiment intensity. It considers degree modifiers—also known as intensifiers or booster words—which can amplify or reduce the perceived sentiment intensity. For instance, phrases like “The service here is extremely good,” “The service here is good,” and “The service here is marginally good” highlight how modifiers impact the sentiment ratings. After reviewing the documentation on the [VADER GitHub](#) page, I imported the necessary libraries and successfully implemented the model. VADER outputs a sentiment dictionary with four values, quantifying the sentiment strength. An example output might be: {‘compound’:

0.5859, 'neg': 0.0, 'neu': 0.513, 'pos': 0.487}, which provides a nuanced view of the emotional tone conveyed in the text.

Code for VADER Sentiment Analysis model:

https://github.com/ojas-chaturvedi/NLP-Gun-Legislation/blob/master/sentiment_analysis/vader_model.py

In the upcoming week, I plan to add at least 2 more Rule-based sentiment analysis models to my project to diversify the methodologies I'm applying. This expansion will help me compare their performance and effectiveness in interpreting legislative texts, offering a broader perspective on their respective strengths and limitations. By integrating a variety of models, I aim to ensure a more robust and comprehensive analysis of sentiment within the data. Additionally, using multiple models serves as a form of cross-validation, helping to mitigate biases that might be inherent in any single model's methodology. This approach enhances the reliability of my sentiment analysis results. Once I complete the implementation of the Rule-based sentiment analysis models, I will transition to exploring Machine Learning and Transformer models. My goal is to have at least three of each type to ensure the most accurate and comprehensive results possible. Concurrently, I will begin drafting the Literature Review and Methodology sections of my research paper, with the aim of completing these sections in preparation for the final paper due on April 29th. Additionally, I am preparing a presentation to share my progress and receive feedback during a rehearsal with my Teacher Advisor, Dr. Travis May, this Thursday. This presentation will be crucial for refining my approach and gaining valuable insights as I continue to advance my research. As usual, I will continue to fulfill my responsibilities in my internship and AP Research class.