COMP 4601

“Wing It” Project Report

By Avery Vine (100999500) and Maxim Kuzmenko (101002578)

ABSTRACT

Wing It is a web application that provides a balanced news diet. A user provides a news article, and then the application determines where the content of the article is positioned on the political spectrum, and provides relevant articles of similar topics classified by political categories for consumption at the user’s leisure. In the era of political divisiveness and increased polarization, *Wing It* brings users analytics that can give them more insight about their political choices, and gives them the tools and sources to view and observe politics from a more objective viewpoint.

INTRODUCTION

In today’s day and age, journalism caters to their target demographic in order to push a political agenda, as opposed to providing readers with a neutral perspective on current events. This was the motivation behind the creation of *Wing It*: we want to provide users with a way of determining whether the publication they are reading leans towards one end of the political spectrum or the other, and offer them access to alternative news articles that offer different points of view.

Following this introduction, there are six sections to this project report. Firstly, any background information required to understand this project is outlined in the Background section. Following that, Related Work informs you of similar documentation/projects that deal with similar issues (i.e. other ways to solve the problem). Thirdly, the Methodology section provides a detailed overview of the requirements we had going in, our approach to solving the problem, and our choice of algorithms. In the Discussion section that follows, we talk about choices that ended up being successful in the long run and pinpoint things that caused us issues. Finally, the project is summarized in the Conclusion and Future Work section, where we discuss future possibilities for this project.

BACKGROUND

In order to understand some of the following technical details, the reader should ensure that they are familiar with the Naive Bayes algorithm, which is used in the context of determine political bias of a body of text:
https://en.wikipedia.org/wiki/Naive_Bayes_classifier

RELATED WORK

There are several similar projects that show the political leaning of a certain publication. However, it it much harder to find one that shows the political leaning of a given article. An example of the former would be:
https://mediabiasfactcheck.com

As an extension to this project, the Stanford NLP API could have been very useful to use in determining the class sentiment values with a higher rate of success:
https://stanfordnlp.github.io/CoreNLP/

METHODOLOGY

To create analytics for users that may give them more insight about their political choices, we needed to create an analytics framework that would best represent the main objective of the application: to make the user view the content of an article from a more objective perspective, and more broadly, view political events more objectively.

Due to the limited time constraints, we decided to set our requirements to involve only analysis of the political leaning of a provided article and simple (algorithmically naive) recommendation of other articles. In the end, we settled on using the Naive Bayes algorithm to determine how words in a given article related to political leanings. More on our choice of recommendation algorithm can be found below, as well as in the Discussion section.

Using Naive Bayes to calculate political leanings proved to be a simple yet effective implementation of the first portion of our project. We started off by providing the algorithm with lists of online publications that historically lend themselves to one of our three political leanings (left, centrist, and right). From there, common words were collected and the probability of each word appearing in a body of text belonging to a particular class was calculated, concluding the training portion of the algorithm. As a result, when a new article is passed into the algorithm, the contents would be extracted and compared to each class, providing a score relating to where the article would be positioned on the political spectrum.

During the implementation, we decided to not only to represent the political class an article falls upon, but also have a gauge/spectrum representing the political “values” of the article: values closer to 0 represent more left-wing views, values closer to 50 represent centrist views, and values closer to 100 represent right-wing views.

To do this, we used our sentiment analyzer to generate sentiment values for each of the political classes, which alone could determine what political class the article is leaning towards. However, we needed to take it further. We needed not only to do comparisons of the sentiment values, but also to calculate a single figure that is between 0 and 100 that is the representation of all of the class sentiment values.

This was not too difficult. Our steps were:

1. Scale the total of all of the class sentiment values to 1.0 with scalar X
2. Divide all of the class sentiment values by scalar X, then set each of the class sentiment value as the log base 10 of the amount of leading zeros for that class sentiment.
3. Adjust all of the class sentiment values by dividing each of them by the sum of all class sentiment values; thus the sum of all the sentiment values should add up to 1.0.
4. Initialize the leanAmount to 0.5, and for each class sentiment value, add/subtract to it the class sentiment value multiplied by the scalar, which is the reciprocal of the number of class sentiments. The adding or subtracting operation would depend on the index of the class sentiment value in the list; the higher the index, the more right it would lean, the smaller the index, the more left it would lean.
5. This resulting sum, leanAmount, after multiplying it by 100, would be between 0 and 100 and would be the representation of all of the class sentiment values.

While our leanAmount figures were mostly correct, they suffered from the flaw that was the inevitable side-effect of scaling extremely small numbers with large differences between them. More covered in the discussion section.

DISCUSSION

If there was one thing to be learned during our experience working with real-world data to extrapolate meaning: it is that is really difficult to make sense of all the data that is out there. Even with large sample sizes, articles often came back from historically left or right wing sources as having a political leaning of exactly the opposite of what would be expected.

There are various things we could do in the future if time and resources were more plentiful. One is to include older data sources for more timely consistency; if too narrow of a time range is chosen, the political view may be associated with one or a small set of events that may not define the political position as a whole. For example, choosing articles for training data just from the time period of November 2016 for either left or right wing sources may be wholly misleading, as there was an American Presidential election at that time with many similar topics and there were very polarizing views between the two candidates. Taking a larger set of articles as training data for a political class that evenly distributes the articles across time would decrease the importance of certain events happening in specific periods of time and would evenly distribute events, resulting in a more accurate training set of documents for the political class.

Another potential improvement would be move away from supervised training. Since we had to manually tell the system which publications are left and right wing, it ensured that the system would be fundamentally flawed with whatever biases we attached during the training stage. If we could group articles by political leaning without the need for human interaction, it would greatly improve the reliability and trustworthiness of the system.

Continued from the Methodology section with regard to class sentiment values, there is a certain issue that makes values tend towards specific category position on the political scale.
This issue is the logarithmic unevenness caused by the lack of accounting for the tiny class sentiment values (Ex.; E^-3000) in the graph. This quickly becomes more complicated as the class sentiment values may differ from each in terms of being E^100 times more prevalent than other values.

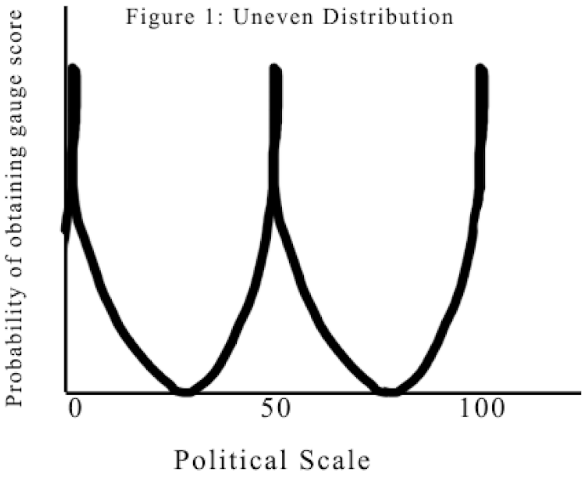
Figure 1: Uneven Distribution

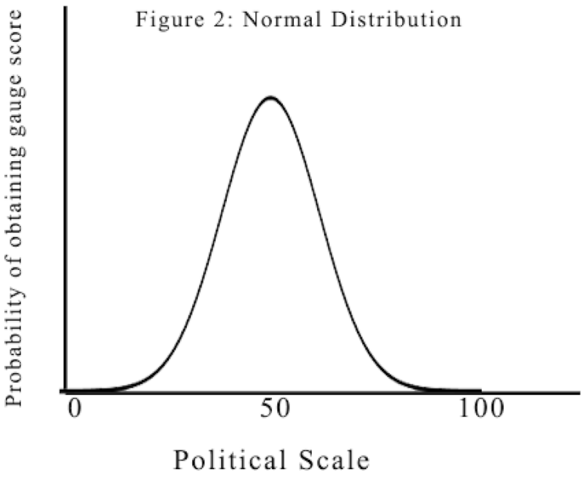
Figure 2: Normal Distribution

As an example, the class sentiment values for left, centrist and right are respectfully 2 x E^-2523, 5 x E^-2526, and 6 x E^-2000. An obvious error arises: when attempting to do step 3, as noted in the methodology section: the values would often time leave one value that is much higher than the other sentiment values, in this example, scaling would these values would make the right sentiment value take up roughly 99.9999998 out of 100 points, while the two other class sentiment values take up a very tiny percentage. This example is shown in Figure 1: Uneven Distribution. While the article in this case leans much more towards a right-wing view, it is crucial to find a method to "flatten" these values so that a more accurate representation of the view is true; In this case, the chance of a certain article to land in and be 99.9999998% republican should be considered to a very rare case and should follow a normal distribution, as shown in Figure 2: Normal Distribution.

As a result, the amount of political leaning from a scale of 0 to 100 did not differ for the categories shown. For the left-wing categories, we typically see a range from 23 - 27 points for political leaning, for the centrist category a range of 46 - 53, and a range of 72 - 78 for the right category.

One way that we helped even out the class sentiment values could be to take the $\log_{10}$ of the amount of trailing zeros of each sentiment value; essentially replacing the sentiment values with their respective number of decimal places, scaled in a logarithmic fashion. This allows for more even and more distributed values, however as we saw in our tests, there needed to be some more evening involved.

In summary on this issue, since the class sentiment values could potentially be very radically different for each different article, approximating the correct gauge value would be very difficult.

CONCLUSIONS AND FUTURE WORK

In the future, there are many improvements that we would like to make to this project. Implementing a proper recommendation algorithm would be a good place to start. The one we created, while it got the job done, was by no means a mathematically sound or industry approved algorithm, and there are many alternatives that we could have used. In addition, another potential goal for the recommender in particular would be to give it the capability to recommend the exact same article from another publisher with a different political leaning. This would ensure that the user is getting the most balanced news diet possible.

In addition to using a new algorithm for recommendation in general, it would be incredibly helpful to be able to take into account calculating political leaning not just the article in question, but the political biases of the article's author and the historical biases of the publication. This would allow us to build a comprehensive profile that could tell us if a particular article is an outlier, or if maybe our preconceived notions of certain publications' biases were incorrect.

Another area in which we could improve the program would be the processing of text during training and analysis. As we only used stop word filtering, we let simple mistakes through such as multiple conjugations of the same word appearing in the database. We would ideally want to implement stemming/lemmatization, as well as add the ability to take into account negation in order to get a better understanding of more complicated blocks of text.

Finally, we could improve the program by storing the articles that users search for, and using them as part of the recommendations for other users in the future.