Latent Probabilistic Model of News Sources

Project Alpha Prototype Report

Army Cyber Institute

Machine Learning for Media Bias
William Hiatt, Gabriel Matthew, and Deven Biehler
02/21/2023

TABLE OF CONTENTS

I. Introduction

This document serves as a guide to the Latent Probabilistic Model of News Sources. It will present new and updated details of the current state of the project as well as the direction the project is going. It will include information such as the social science aspects of the project, the open source models we will be using, models we will be creating, research that we will be using and research we will be doing internally, and any requirements that we currently have. Finally this document will quickly go over the bio's of group members on the team that are working on developing the model as well as all the stakeholders that the project will effect.

I.1. Project Introduction

The US Army Cyber Institute needs a model used for evaluating bias within individual news articles. This model will be used to give a bias analysis as well as topic analysis on a specific news article. The model will grade the article on a scale based on the different biases that the model looks for. With this scale the Army will be able to fight information warfare more efficiently, both on US based news sites and foreign news sites. The end goal of this project is to create a new model for the Army that will rate individual news articles on their bias and misinformation. The model will establish an ecosystem of news articles and sources that will display biases and sentiment values based on the major topics within each article. This ecosystem will be displayed graphically for the user to make insights about the data. The model will use a mixture of currently available software as well as software created by the team.

I.2. Background and Related Work

Concerns about media bias and its potential impact on society have grown in recent years. Misinformation, polarization, and even discrimination can result from biased news. This will specifically be used by media analysts in the US Army to predict the ecosystem of the media.

Due to large amounts of misleading information or complete misinformation, the US military has grown increasingly interested in fighting back. With their help, we are hoping to build a model to decrease the amount of misinformation and the speed at which it spreads.

There are some sites that currently provide a similar service such as allsides.com, mediabiasfactcheck.com, and adfontesmedia.com. These sites currently just look at the media outlet instead of breaking down individual articles. These outlets also tend to use humans to help determine this bias and that brings in another potential bias.

I.3. Project Overview

The media has a huge duty in the form of distributing news to the American public. The information plays a huge role in how people vote. The current offerings in finding media bias aren't as detailed as they should be and still contain a bias in themselves as it's often a human who is assigning these bias's. For example, allsides.com has CNN ranked as far left while adsfontesmedia.com has them as slightly left leaning. This is just a single example, but it shows that there is even a bias within deciding bias. In addition one post from a media outlet could contain no bias and be factually correct while another may contain a lot of bias and not be factually correct. The goal of the team is to combat this by creating a model to find the bias, without any hint of human bias, of a single news article in regards to a specific topic.

The team, with the help of mentors at the Army Cyber Institute, will create a new model that will address the issues presented in current media bias tools. Through research, designing, building, and testing the team will deliver a media bias tool to help fight information warfare.

The project is being built from the ground up. There are no limitations on how we create the model. The team has decided to use Python to create the model. Python is extremely popular for machine learning and will be a great platform to use. The code will be housed on GitHub, this allows for easy version control and collaboration.

Using Python allows for the use of some popular tools. Some tools the team will be using include, spaCy for natural language processing, textBlob for sentiment analysis, Pandas for managing data, and Beautiful Soup for webscrapping. These tools will all be used in conjunction to create a model that fits the needs of the client.

The team is also in contact with two Social Scientists at Army Cyber Institute to help in finding the most important bias's to use as well as the best way to display the output. The team will be using sentiment analysis and fact selection as the primary drivers in deciding a bias. As the project progresses, we plan to include additional secondary drivers.

I.4. Client and Stakeholder Identification and Preferences

Our client is the US Army Cyber Institute with Senior Research Scientist Iain Cruickshank as our mentor and primary contact for the project. The product will be used and maintained by the Army. There are several stakeholders within the Army Cyber Institute including Iain and his colleagues.

II. Team Members - Bios and Project Roles

William Hiatt is a 4th year software engineering student at Washington State University. His skills include C#, Python, SQL, Java, C++, and GoLang. He has prior experience working as a software engineer intern at Kochava as well as a junior web developer at Washington State University. For this project, William will act as a team lead, main developer and designer of the model, and work with his team to create a viable machine learning model.

Gabriel Sams is a 4th year Computer Science major at Washington State University. His skills include C#, Python, Database development/management, SQL, machine learning, software development, agile process, test-driven development, and data science. His prior experience includes mobile application development, database development and management, and unit, end-to-end, and functional testing. For this project, Gabriel will act as a main developer and designer of the model, and work with his team to create a viable machine learning model.

Deven Biehler is a 4th year Computer Science major at Washington State University. His skills include C/C++, Python, SQL, machine learning, HTML/CSS, agile process, and data science. For this project, Deven will act as a main developer and designer of the model, and work with his team to create a viable machine learning model.

III. Project Requirements

III.1. Spike Stories

III.1.1

Determine the tools to be used for sentiment analysis/textual analysis. Users will be able to view per-article data gathered from NLP and sentiment analysis. The data will be clearly displayed and comprehensible so the user can interpret it in a professional setting. Our team will need to determine open-source tools that will provide the highest accuracy and reliability to our data.

III.1.2

Develop a model at every step of document and ecosystem analysis to build a usable model of media bias detection. The user will be able to gather useful data from three compounding levels of analysis. The tool itself will focus on displaying the “news ecosystem” for users, so the previous layers need to provide useful data to construct it. We will explore and develop algorithms to compile meaningful results at every level of news bias modeling.

III.1.2.1

Create the base model for our software, that takes a single document and provides data on its sentiment of topics within the text as well as factuality of the documents. This model will be used to build a larger corpus of data for many separate documents.

III.1.2.2

Develop a model that compares single documents to each other to find matching topics, and sentiment on a per-article basis. These comparisons will be stored to create a larger ecosystem of article comparisons.

III.1.2.3

Building from the per-article model, create a model that compares and displays the overall ecosystem of articles and how they relate to each other on a certain topic. The model will compare sentiment and factuality based on the models’ textual sentiment analysis and group them/sort them in a meaningful way.

III.1.2.4

The final model must be able to categorize and sort the articles to show meaningful relationships between articles that match topics. Develop an algorithm to meaningfully sort the article ecosystem into a graph that is comprehensible to the users.

III.1.3

Find the most meaningful ways of measuring bias in text. Users will need the most accurate bias data for each topic. By researching and talking with social science professionals the team will create the most effective way at measuring bias so that the users have the most accurate information available. In the beginning only a couple different forms of bias will be evaluated with plans to expand in the future.

III.1.4

Explore methods of fact-checking as well as opinion recognition. Users will need to see when an article is falsifying data or otherwise lying to promote a bias. Fact-checking is an essential point of data when determining the motives of a particular article or source. We want to research possible ways of assessing the validity of a fact. Fact-checking will allow us to determine if a piece tries to persuade a user into a specific worldview. If we can find the lies, we can detect future lies that could be told in articles yet to be published.

III.1.5

Determine tools to be used for topic modeling. In order to determine the bias per topic, we need a model to extract the topics from the articles. The topics will be extracted from each article and displayed each with a range of sentiment, ranging from positive or negative toward the topic.

III.2. Functional Requirements

III.2.1. Sentiment Analysis

Per Topic: Using a pre-trained natural language processing model, we receive an emotionality score towards an array of topics. Sentiment analysis is crucial to predict the emotional bias towards a specific topic.
Source: Senior Program Manager with Army Cyber Institute originated this requirement. The requirement is necessary for the bias analysis.
Priority: Priority Level 2: Essential and required functionality

Per Article: We receive an emotionality score towards the entire article using a pre-trained natural language processing model. The goal is to give the model more crucial data about the article's topic.
Source: Senior Program Manager with Army Cyber Institute originated this requirement. The requirement is necessary for the bias analysis.
Priority: Priority Level 2: Essential and required functionality

Per Source: Using a custom-built machine learning model, we can determine where the source will lay in the entire ecosystem of media news sources. The goal is to allow a user to understand the worldview of a source within the media ecosystem.
Source: Senior Program Manager with Army Cyber Institute originated this requirement. The requirement is necessary for the bias analysis.
Priority: Priority Level 2: Essential and required functionality

III.2.2. Topic Modeling

Modeling Relevant Topics: Determining the topics is essential to model a per-topic bias analysis. The model learns bias towards an array of topics to portray the media ecosystem to the end user.
Source: Senior Program Manager with Army Cyber Institute originated this requirement. The requirement is necessary for the bias analysis.
Priority: Priority Level 2: Essential and required functionality

III.2.3. Similarity Analysis

Similar Article Analysis: When learning the bias of specific sources, we can use their similarities to other sources to predict missing values. The model will look for any blatant copying from sources and take it as valuable data to make accurate predictions.
Source: Senior Program Manager with Army Cyber Institute originated this requirement. The requirement is necessary for the bias analysis.
Priority: Priority Level 1: Required for an accurate model

III.3. Non-Functional Requirements

Reliability/Accuracy

This tool should most importantly provide accurate and reliable information based on the input given. The tool should provide the same results for the same input every time, as well as provide data that is usable in a professional news media environment.

Extensibility

This tool will be extensible to other domains within online media, including other languages and other political environments. While the tool itself will only work for English language and American news sources, it will allow for the insertion of new tools and libraries to adapt it to these new domains as its users see fit.

Usability

This tool will aim to be usable by industry professionals and those who have domain knowledge of American news, but not necessarily any knowledge of Computer Science or how the tool functions in terms of programming. This means that the tool will include an instruction manual or other documentation that helps users work with the tool. The tool will be designed in an intuitive way.

Content-Focused

This tool will focus on analyzing the actual content of news articles rather than the context surrounding them. The tool will analyze the text itself as well as relevant data and sources to make its calculations–it will not focus on the author, the timing of the article, or other contexts involving the news piece being analyzed.

Comprehension

This tool will provide highly comprehensible data that can be interpreted by industry professionals to make professional decisions in a business setting. It will provide any relevant data to the user and will document any unreliable data or data that could be up to interpretation. Our goal is to be as transparent as possible about the reliability and usability of the data.

IV. Solution Approach

Our solution consists mainly of three models that build upon each other to create a final, working model of the entire problem scope. Each model will take input from the previous model or from inputted data, and provide some interpretation of that data to pass into the next model or to present to the user.

The first model will take URLs of articles in as our input, web scrape the text from the URL, and perform sentiment analysis and topic modeling on the text of the article. This will gather data on the topics the article covers, as well as the bias presented within the article. We think this is a good choice as a foundation because we will need to analyze each article individually if we want to make comparisons between articles and make meaningful connections between them.

The next model will take the data given by our first and create relationships between models based on that data. It will make comparisons of topics, as well as the sentiment between articles. We think this is a solid next step because the data will be passed through to our third article which will organize and display the data. This model acts as a factory between the raw analysis performed and the display of our organized data.

Finally, our third model will organize and represent the data graphically such that the user can interpret the relationships and data provided and use it in a professional setting. It will take the given relationships as well as the single-article data and organize it based on topic, then group articles that are similar and have similar bias. We think this is a good final stage because we will have all of the data from our previous two models which is what we need to show the user, but no way to display it in a comprehensible way. By creating a final model that primarily focuses on organizing and displaying the relationships, we can separate the creation of meaningful data from the displaying of the data and improve extensibility.

IV.1. Model Input

The model will take in URLs of news articles in the form of a list, and will need to scrape the data from each URL to gather the title, author, and text of the article. The specific input will be a single news article that the model will run on. We think this is the best choice of data input due to a URL being simple to gather and run in the model, and that analyzing a single article at a time will make it easier to compile the data for many articles.

IV.2. Data Scraping/ Cleaning

The model first uses Pandas to read in the URL’s provided in the CSV file passed into the model. As it’s reading in the URL’s it then pushed them onto a list. After all URL’s have been read the list is then used in a loop. Each URL int he list is then run through Beautiful Soup to scrape the web page and store the text. The text is then stripped of any white spaces. Then the data is loaded into spaCy. SpaCy then uses textBlob to run the sentiment analysis and returns a polarity sentiment score. We then give the sentiment a positive or negative label, positive for scores over 0 and negative for scores under. After it has looped through every URL it then uses Pandas to put it back into a CSV file with columns for Sentiment Score, Sentiment Label, Positive Words and Negative Words. This process allows for easy usability as the user only needs to input the URL’s of the news articles without having to worry about scraping the data themselves.

IV.3. Sentiment Analysis

We will be using Spacy as our Sentiment analysis tool, as well as the Textblob library within Spacy. The model will take in the article’s text and perform an overall sentiment analysis on the article. It will also perform topic modeling and localized sentiment analysis on those topics to find the sentiment around certain topics within the article. This will be necessary to feed our model the topics and sentiment analysis per article to create a larger ecosystem of articles. We chose these tools because of their ease of use, and public access.

IV.4. Topic Analysis

We will use Gensim, a Python package, for topic analysis. Our model needs to extract topics from each article to be able to categorize it within our media ecosystem. Each outlet will be judged by the model on the topics extracted. We picked Gensim because of its efficient implementation, and because it provides several topic modeling algorithms that can be customized with various parameters to optimize for particular use cases. This adaptability can be beneficial in detecting news bias, where various topics or biases may necessitate different modeling strategies. Beyond that, it integrates well with other NLP tools, SpaCy in particular.

IV.5. Model Output

A visualization of the media ecosystem will be created as the model output. It will represent every article as a node on a graph, displaying the relationships between articles on a certain topic. The per-article analysis will also be viewable for each article. The graph will be sorted by topic, with each source and article being displayed based on sentiment and factuality. The graph will display similarity between articles as distance (closer together means the articles share similar topics.) There will also be links between articles that show similarity in text and sentiment. This will provide a comprehensive and understandable format of the data for our users. We chose this solution approach because the model needs to be understandable by industry professionals who don’t program.