Project 2-2 Propaganda Detection Draft Report Group 03

Emre Pelzer[1], **Oliver van Sonsbeeck**[2], **Mathijs Brouwers** [3], **Andrew Crabtree**[4], **Julius Verschoof**[5] **and Sokratis Hadjichristodoulou**[6]

E–mail: *e.pelzer@student.maastrichtuniversity.nl*[1], *o.vansonsbeeck@student.maastrichtuniversity.nl*[2]
m.brouwers@student.maastrichtuniversity.nl[3], *a.crabtree@student.maastrichtuniversity.nl*[4]
j.verschoof@student.maastrichtuniversity.nl[5], *s.hadjichristodoulou@student.maastrichtuniversity.nl*[6]

ABSTRACT: This study explores the detection of propaganda within digital articles using natural language processing (NLP) techniques. With the pervasive spread of misinformation and propaganda in digital media, there is a pressing need to develop automated systems capable of identifying biased or misleading content. This research focuses on leveraging NLP algorithms to analyze text features indicative of propaganda, aiming to empower readers to discern the underlying motives behind the information presented. Through a modular approach and utilization of the OpenAI API, we aim to develop algorithms capable of classifying articles as propaganda or non-propaganda based on identified features. The study contributes to the ongoing efforts to combat misinformation and foster critical thinking among news consumers in the digital age.

Key words. Natural language processing, Report, Propaganda evaluators, Neural network

Contents

1. MOTIVATION & PROBLEM STATEMENT

News and information are widely available across the internet, and a vast majority of people get their information from digital devices. In 2020, a study found that over 80% of Americans now rely on digital platforms for news consumption over traditional sources like television, radio, or print media (Pew Research Center, 2021). However, the ease of access to this abundance of information comes with a significant caveat: the susceptibility to misinformation and propaganda.

The Oxford Dictionary defines propaganda as "ideas or statements that may be false or present only one side of an argument that are used in order to gain support for a political leader, party, etc." Propaganda is often used in numerous contexts. These contexts can be advertising, political campaigns, social movements and other influencing public opinion on various issues. Successfully used propaganda can control real-world events such as elections and war campaigns. Employing techniques such as emotional appeals, loaded language, and testimonials, propaganda permeates various forms of media, from written text to speeches and visual imagery.

Propaganda can lead to long-term effects of radicalization, social division and erosion of trust. Vulnerable demographics, such as the youth and elderly, are particularly susceptible to its manipulative tactics due to their limited critical thinking skills and lesser understanding of the subjects discussed. Young people may lack the experience and education to critically evaluate information, while the elderly may be less familiar with modern digital media and its potential for spreading misinformation. This raises ethical concerns regarding the legality of fake news dissemination and the use of propaganda. Therefore, it is imperative to develop mechanisms to detect and highlight the presence of propaganda within articles.

To address this, We propose using natural language processing techniques to develop an automated system for detecting propaganda within articles. We will be focusing on detecting specific propaganda techniques within fake news articles. Fake news refers to false or misleading information presented as news. Our model will identify propaganda techniques within the text, especially in the case of articles deemed as fake news. By flagging these techniques, we hope to empower readers to discern the underlying motives behind the information presented.

2. RESEARCH QUESTIONS

- How do we convert propaganda techniques into a list of text features we can extract from a text?
- How successful are these features at classifying articles as propaganda or not?
- Which of these features are most significant to determining the classification of an article?

3. CHOSEN APPROACHES

In order to address the first research question we decided that we would take a list of techniques that apply to propaganda and turn them into a list of features we would then use to identify in text passages. Based on the limited research available, we decided on features such as emotionally charged language, high risk words, positive generalities and stereotypes. The idea being that we would gather a large number of features that an algorithm would be able to find examples of in text in order to assign them some form of score for their frequency.

To address the second research question, we would like to create a modular system that is able to scan for the usage of each of the many features we choose to focus on and assign them a kind of score based on how much of that feature is found. We want to then use these algorithms to test how well they are able to classify documents as propaganda or not by focusing on each of these features individually. In order to produce these algorithms we will use a few approaches depending on their complexity. For simpler modules such as high risk words, we feel as though a simple algorithm capable of finding words in a text would be sufficient, with perhaps the usage of the OpenAI API's assistance to help generate our list of words to search for. However for more complex features, such as ones where the context for words usage may be necessary we want to use the OpenAI API in order to identify these features. We want to experiment with prompting to find out how we can best get the AI to identify correct examples of specific propaganda features within the texts we give it.

Finally, to answer the third question, we will create an artificial neural network to predict if a text contains propaganda by using the scores of all tested propaganda features as input. This way, we would be able to see how well combining the findings from all out chosen features are able to classify documents. Additionally, by doing this we would have the ability to create new neural networks with less features to determine the impact the removed features have on determining the final outcome. By observing how the effectiveness of the classification changes with less features we can get an idea as to which features are more important in the classification.

4. IMPLEMENTATION

4.1. Features

Propaganda can be broken down into many techniques. Given the time frame for Phase Two of our project and our technical ability, we focused on six techniques. We designed evaluators to identify the respective techniques within the text and output a grade showcasing the presence of that technique. The individual features might not be reliable indicators of propaganda on their own. In fact a single feature would definitely be a poor indicator as propaganda come in many forms and can be hard to detect. However, we hope that together the features can give a good representation of the text that the ANN can use to accurately detect propaganda.

If the input is or contains a URL, a simple web scraping module locates and processes the web page's HTTP content, returning it as a text file. Based on how the URL was inputted, this text file is either treated as the text to be graded, or as an addition to the text within which we found the URL.

4.1.1. Emotionally charged language

We anticipated the emotionally charged language module to be more complex than others. This was because we wanted to find instances of emotional charge that weren't tied down to the presence of individual words in certain sentences, but the meaning of sentences as a whole. Because of this, we wanted to attempt something different, so for this module we decided to implement the OpenAI API. This module works by connecting to gpt-3.5-turbo and providing it instructions and a text passage to analyze each time. It is prompted to identify and extract exact sentences that contain examples of emotionally charged language. The response from GPT is compiled into a file, which is later used to create a score between 0 and 1, representing the ratio of sentences containing emotionally charged language to the total number of sentences in the text.

4.1.2. High risk words

Propaganda can be characterized by its content of ´risk words´. These ´risk words´ can be defined as certain words which carry a great deal of power and can be used to influence the readers. For the evaluators, three different kinds of ´risk words´ have been used. The selected types of language are: loaded language, emotional language, and bandwagon language. Loaded language is the broadest type of 'risk words'. It includes a broad range of words, from racist and politically charged words to swear words. (Wikipedia 2024).

The emotional language contains words that try to evoke strong emotions from the reader. It uses words such as 'hate' or 'love'. This list has been taken from various sources (Centervention n.d.) and from Large Language Models which generated parts of the emotionally charged words.

Finally, the bandwagon language consists of words that tries to give the reader the feeling of unity. It tries to give the reader the feeling that they need to choose a side, and that the reader should join the writers side. This can be words like 'come together' or 'unity'.

Each type of 'risk words' has its own evaluator. The way the evaluators work is as follows: First, the ratio of each type of 'risk words' to the amount of words has to be determined for both the propaganda dataset and the non-propaganda dataset. After doing this, the propaganda and non-propaganda dataset will get ratios of the amount of 'risk words' are in both the datasets. This gives us a range of ratios. For example, if the ratio of 'risk words' to total words in the propaganda dataset is 0.01 and in the non-propaganda dataset is 0.001. This now gives a range. Now, if a new text file has to be evaluated, the evaluator will count the instances of 'risk words' in the text. Once it has enumerated the amount of 'risk words', it will get a ratio of risk words. This ratio will now be compared to the pre-computed range. For example, if a new text file has a ratio of 0.005 'risk words' to amount of words in the text file, the likelihood of the text file being propaganda is 0.50 according to this evaluator.

4.1.3. Positive generalities

Propaganda can also be characterized by positive generalities, also referred to as glittering generalities. This technique consists of using very positive words without actually saying anything of value to support the positive claims. Saying things like "amazing", "wonderful", and "excellent" but overall not supporting this with concrete examples. To detect this technique we search for words with positive sentiment. We then divide the number of these words by the total length of the text. We assume that the ratio of positive sentiment words to the total length of the text can indicate whether the positive words are supported by evidence. If the ratio is high it means that there would not be a lot of words in the text which can support the claims and therefor has a higher chance of being a positive generality. The output score is from 0 to 1, and the score is the ratio slightly modified to ensure that mid level scores like 0.5 get boosted to be a bit close to 1.

4.1.4. Stereotyping

Another common feature within propaganda is stereotyping. Stereotyping is a technique where a person or group is oversimplified and often given a negative label. This is done to present these people in an inaccurate and oversimplified manner, invoking negative emotions towards them. To detect stereotyping, we use the OpenAI API, which works very similarly to our emotionally charge language method evaluators. We prompt it to identify and extract instances of stereotypes within a given text. Using this we create a score between 0 and 1 of a ratio between the total sentences and the sentences which GPT has identified as containing a stereotype.

4.2. Neural network

Most of our functionality comes from the creation of our artificial neural network. It has an input layer of 6 neurons to match the number of features in our vectors, followed by a dense layer of 64 neurons using the ReLU activation function. ReLu was chosen due to its increased efficiency when compared to other functions. The final layer is a single neuron using a sigmoid activation function. Both of these choices were made to make this artificial neural network work as a binary classification for propaganda. For the same reason, the artificial neural network uses binary cross-entropy loss to calculate error. Finally, it uses the Adam optimizer for its adaptive learning rate capabilities.

4.3. Future features

Our neural network takes the aforementioned features to produce a final propaganda score. These 6 features are not final, and there are additional features that can be used to identify propaganda. For this reason we created additional evaluators for different features. While the ANN does not use these features right now; we added implementation to detect additional features which could be incorporated in the future.

4.3.1. Cherry Picking

Cherry picking is 'the action or practice of choosing and taking only the most beneficial or profitable items, opportunities, etc., from what is available'Cambridge Dictionary (2024). This is a also a feature of propaganda as one side will usually only select the information that fits their narrative. To detect it our evaluator uses OpenAI API. First we use it to create a summary of a given text. Then it performs a google search with that summary and collects the top 10 search results. After which we use the Open API again to give us a summary of these 10 results. It then compares the vector embedding of the top 10 summaries with the original summary. If the top 10 have very similar embedding we would say that it is not cherry picking, and if they have very different embedding we would say that it does contain cherry picking. The final score is calculated by taking one minus the average cosine similarity between the summary and the titles. This is to ensure we have consistency with the other evaluators and that 0 means no cherry picking and 1 means it has cherry picking.

4.3.2. Sentiment Evaluators

We also discussed he idea of adding very basic evaluators to give the ANN as much data about the text as possible. For this reason we created two sentiment evaluators: one for positive sentiment and one for negative sentiment. They work the same way but look for the opposite. Each sentiment evaluator keeps track of either positive sentiment or negative sentiment depending on which it is. It then creates a score of that sentiment by dividing it by the length of the text. This final score is between 1 and 0, where 1 corresponds to a lot of the given sentiment and 0 none.

4.4. Data processing

We created the file dataset.py to process our acquired data into appropriate vectors for training and testing. This file iterates over the files in both the propaganda and non-propaganda folders alternately, feeding each file through the evaluators to compile all the scores into vectors. The function was designed to allow for more data to be processed at any time. Whenever it is run, files are created containing checkpoints for the last document in either of the folders that was processed before the last termination. These checkpoints allow the file to resume processing from where it left off by iterating over all previously processed texts. Additionally, it contains a save and load feature that allows it to add any new processed data and aggregate it with any that were already completed.

5. EXPERIMENTS

To determine how well our list of features can be used to classify documents based on their propaganda content we experimented using our artificial neural network trained on our list of features. This experiment was conducted using only a subset of our entire dataset as the processing of each text passage into vectors of the features we chose not only takes lots of time but also has a monetary impact on us due to our usage of the OpenAI. For this reason, our entire dataset for the artificial neural network contains 2500 vectors with an even split between propaganda-filled, and not propaganda-filled texts. Once the data has been processed the vectors of features and the vector containing their labels are temporarily merged to reorder the data using numpy's shuffle feature (Numpy

2024). This feature will randomly shuffle the order. This was done because due to how processing occurs, the vectors are ordered in the same way as their corresponding documents were in the original dataset (Peutz 2022). Once reordered and the labels were separated from the feature vectors, two splits were made within the processed data to produce the training, validation, and test sets. The validation and the test set were made using 20 percent of the processed data, leaving the last 60 percent to the training set. The neural network could then be trained using the training data and validation set. Once complete with training the test set could then be used to compare the true labels of the test set against the predicted labels of the neural network. With these data sets, we were able to perform several experiments to test the effect of variables on the precision and recall of the network.

Firstly when it came to our stereotype evaluator we were considering implementing it using OpenAI like the emotionally charged language evaluator. However, we needed to know if this would improve out ability to classify the documents or not. To test this we created two artificial neural networks, one that used the old processed data where the method for stereotype evaluation was more of a brute-force approach and one that used the new OpenAI API method. Once created both ANN had their corresponding test sets used to find their precision and recall so they could be compared.

Secondly, due to the lack of a true state-of-the-art model for the topic due to this issue being quite niche, we chose to perform a comparison between our ANN and a logistic regression model which would act as our baseline model. This experiment would also address our second research question of how effective our ANN was at identifying articles as propaganda-filled or not. It should also be noted that this experiment and future ones use the new stereotype evaluation method in favour of the old one. For this, a logistic regression model was created using the solver parameter liblinear and random state 0. Once created the model was trained using the same training set as the ANN. Since logistic regression cannot easily incorporate a validation set into its training this set was not used for the logistic regression model. Once trained, the predictions of the logistic regression model were compared with the actual classifications to determine the precision and recall of the model. Similarly the predictions of the ANN were collected to find the precision and recall of the artificial neural network as well.

The final experiment we performed was determining which features are most important to the classification result. We chose to approach this by performing several tests These tests comprised of training six new ANN's where each one was trained on the same data as the original, with one of the features removed. As a result each new ANN can be used to show the impact of each feature on the precision and recall by comparing the changed scores to those when all features are present.

6. RESULTS

Approach	Precision	Recall
Brute Force	0.583	0.539
OpenAI	0.676	0.705

Table 1: Precision and recall of brute force vs OpenAI approach for stereotype evaluator

This table displays the results for the first experiment where we compare the effectiveness of the brute force approach to the stereotype evaluator vs the OpenAI approach. The top row shows the results obtained from the brute force approach while the bottom shows the results from the OpenAI approach. The left column denotes the precision of the ANN and the right column denotes the recall of the ANN.

Model	Precision	Recall
LogReg	0.687	0.682
ANN	0.676	0.705

Table 2: Precision and recall of the logistic regression model compared to the artificial neural network model

This tables shows the recall and precision for the logistic regression model compared to the artificial neural network model. The top row shows the results for the logistic regression while the bottom row shows the results for the artifical neural network. The left columns displayes the precision of the corresponding model while the right column shows the recall.

Feature removed	Precision	Recall
None	0.676	0.705
Emotionally Charged Language	0.626	0.805
Stereotypes	0.683	0.678
Positive Generalities	0.651	0.720
Emotional Language	0.685	0.682
Loaded Language	0.654	0.536
Bandwagon Language	0.709	0.682

Table 3: Precision and recall of the ANN's trained by removing one feature. Red denotes a worse score and green denotes a better score compared to the original ANN.

This table displays the differences in precision and recall for ANNs trained on different removed features. In the left column, the missing feature for the ANN is displayed, the center column shows the precision of the respective models and the right columns show the recall for the respective models. Numbers highlighted in red indicate a decrease in value compared to the original ANN score while one in green indicates an increase compared to the original.

7. DISCUSSION

Discussing the first experiment addressing our first research question, the results, as can be seen in Table 1 show a large increase in precision and recall when observing the OpenAI approach when compared to the brute force approach. This result is unsurprising as OpenAI's large language model would be expected to be better at picking out examples of text features compared to a more crude method that cannot properly discern word relations within a text. Due to these results we can confidently say that more complex features within text should likely be evaluated through an OpenAI approach to be able to classify texts more optimally.

The second experiment which addresses our second research question, shows its results in Table 2. Here we can observe a slight increase in precision in the logistic regression column and a decrease in recall. This seems to imply that the logistic regression model, given a new document, is more likely to classify it as propaganda. This would explain having a higher precision but lower recall. That being said if we focus on the introduction we made for our project we stated that we want this product to be able to inform readers and warn them of potential propaganda within the articles they read. For this reason, we feel that recall is the more important feature between precision and recall, as a high recall ensures that it is less likely for a propaganda article to be classified as non-propaganda. For this reason, we believe the ANN aligns more closely with our goals and thus is the better approach for our classifier.

For the final experiment, we wanted to see which features were most important to our classification which we did by training six ANNs where each one had all the original data but had one feature removed. The results can be seen in Table 3 where every red number shows a worse score in that field when compared to the original ANN, and a green number shows an improved score. From observation, we can see that the removal of loaded language leads to a decrease in both precision and recall with the amount of decrease being more significant for recall. We can conclude from this that the feature provides important data for the ANN to learn to classify better and as such we can surmise that it is one of the most important. For the other features, while their removal does lead to a decrease in either precision or recall, it also leads to an increase in the other which makes us certain they are not as important as loaded language. That being said as mentioned previously in our project we believe good recall is preferable to better precision, and with this in mind, it may be beneficial for us to take into consideration the features that could be removed to increase the recall. These features would be emotionally charged language and positive generalities, while both show a decrease in their precision when removed, the recall for the model goes up. Thus it may be possible that a future version of the classifier may disregard these features entirely to increase its recall score.

8. CONCLUSION

Our goal was to set up an automated system that could detect propaganda to help protect people from many sources of misinformation. Propaganda is a very broad term which made it challenging to work with. We used the most common features of propaganda to help us identify if a certain text contains propaganda. These features combine in a neural network to give a text a final score to determine whether it contains propaganda or not. Our final neural network is a good step in the right direction for such detection, but our results still show room for improvement. We have identified ways to we could attempt to improve our method, such as adding more feature evaluators to give the artificial neural network more information to work with, or removing features so that the recall could increase. In addition to this, it should be taken into account that our ANN had a very limited amount of training data due to time and monetary constraints. T more data would likely lead to further improvement in our approach. Nonetheless, our method already shows promising results and can begin helping detect propaganda to prevent misinformation.

REFERENCES

Cambridge Dictionary. 2024, Definition of "cherry-pick" — Cambridge English Dictionary, accessed: 2024-06-25

Centervention. N.D., List of Emotions: 135 Words that Express Feelings, https://www.centervention.com/list-of-emotions-135-words-that-express-feelings/, accessed: 2024-05-27

IBM. N.D., Tone Analyzer package, OpenWhisk Documentation, retrieved from https://cloud.ibm.com/docs/openwhisk?topic=openwhisk-pkg-tone-analyzer

Incrementors. 2024, Types of Propaganda Used in Advertising, https://www.incrementors.com/blog/types-of-propaganda-used-in-advertising/

Kim, H. 2020, Medium,), https://medium.com/@jihwangk/fine-grained-propaganda-detection-and-classification-with-bert-dfa4ecea321

Kwintessential. N.D., The Language of Propaganda: How Words Can Sway a Nation, retrieved from https://www.kwintessential.co.uk/blog/general-interest/the-language-of-propaganda-how-words-can-sway-a-Nation

Leonard, J. N.D., Propaganda Techniques, retrieved from https://www.uvm.edu//jleonard/AGRI

Mediawijs. N.D., Power of Language as a Propaganda Tool, https://www.mediawijs.be/en/article-overview/power-language-propaganda-tool

Mitchell, T. M. 1997, Machine Learning (McGraw Hill)

News, C. 2021, Words and phrases you may not know are offensive, [Online; accessed 24-June-2024]

NumPy. 2024, numpy.random.shuffle, NumPy, https://numpy.org/doc/stable/reference/random/generated/numpy.ra

OpenAI. N.D., Introduction to the OpenAI Platform, OpenAI Platform Documentation, retrieved from https://platform.openai.com/docs/introduction

Peutz, S. 2022, Misinformation, Fake News, and Propaganda Dataset (79k), https://www.kaggle.com/datasets/stevenpeutz/misinformation-fake-news-text-dataset-79k/data

Pew Research Center. 2021, More than eight-in-ten Americans get news from digital devices, retrieved from https://www.pewresearch.org/short-reads/2021/01/12/more-than-eight-in-ten-americans-get-news-from-digital-devi

Poulter, B. 2001, Propaganda, Eastern Illinois University, retrieved from https://www.ux1.eiu.edu/bpoulter/2001/pdfs/propaganda.pdf

Wikipedia. 2024, List of ethnic slurs, https://en.wikipedia.org/wiki/List_of_ethnic_slurs