Detecting Propaganda via Logical Fallacies in Political Speech

Safiyyah Ahmed
Senior, Computer Science
University of Michigan
Ann Arbor, USA
ahmedsaf@umich.edu

Zachary Eichenberger
Junior, Computer Science
University of Michigan
Ann Arbor, USA
zeichen@umich.edu

Jensen Hwa
Senior, Computer Science
University of Michigan
Ann Arbor, USA
hwaj@umich.edu

I. Introduction

Logical fallacy detection is a task of significant importance in natural language processing, with applications to real world issues. Our societies exist in a world where large volumes of written, spoken, and video information can be shared on a global scale via social media, news networks, and other forms of electronic media. Unfortunately, today's social and political landscapes suffer from an increase in biased misinformation. NLP models can be used to detect logical fallacies and be incorporated into a larger pipeline for fact-checking and preventing the spread of disinformation.

Propaganda and logical fallacies are often interrelated. Indeed, the rhetorical techniques used in propaganda are often closely related to logically fallacious arguments [**?**]. Significant recent work has been done on fallacy detection, including the SemEval 2020 task 11, "Detection of Propaganda Techniques in News Articles" [**?**].

In this project, our objective is to create a machine learning model to detect propaganda in news articles and political speech. More specifically, we seek to leverage prior work in detecting logical fallacies to create a textual-based model for propaganda detection.

We first train a model to predict types of logical fallacies given examples from educational websites. Then, we transfer this to a related task of predicting whether or not given news snippets are of 'propagandistic' content. Finally, we apply this model to novel sentences taken from presidential speech to qualitatively evaluate what the model has learned.

Each of the authors made a roughly equal contribution to the project. Safiyyah completed the pretraining step on the logical fallacy examples dataset, Zachary worked on creating models using precomputed Bert embeddings and comparing the models to naive techniques, and Jensen led the work of creating and transferring the model to the presidential speeches dataset.

II. Related Work

Logical fallacy detection is an area that has seen significant academic focus in recent years.

One approach to fallacy detection [**?**] is to use zero-shot learning on large-scale transformers such as RoBERTa and BART in order to train their capability for detecting logical fallacies. More specifically, in the training process, existing NLP tools such as entity detectors and toxic language detectors were used to identify class-specific signature features of logical fallacy types such as *appeal to authority* and *ad hominem* attacks. Other logical fallacies were detected by checking relations of multiple spans in order to identify logical errors caused by confusing correlation with causation, such as in *circular arguments, fallacy of relevance, etc*. These methods were applied to detecting logical fallacies in climate change articles.

Additionally, the SemEval 2020 task 11 focused on classifying logically fallacious propaganda techniques in documents *known to contain propaganda*. It drew submissions from at least 44 teams. Top performing teams used language models like BERT to create embeddings, followed by a sequence model, often a bidirectional LSTM for the final classification task [**?**].

Conceptually, the use of logical fallacies also have some intersections with propaganda techniques. An approach to detecting propaganda [**?**] is to use a Maximum Entropy classifier and focus on the effectiveness of the different word representations: word n-grams, lexicon, vocabulary richness, readability, and character n-grams in a main 2-way classification task: propaganda vs. non-propaganda.

Our project provides a novel approach by widening the scope of logical fallacy detection to encompass instances of propaganda in political and social news articles. We leverage two distinct training datasets, one consisting of labelled logical fallacy examples and the other consisting of news articles labelled as propaganda or not. We draw from the conclusions from the SemEval task by using the same transformer-based models for classifying logical fallacies, pretaining them on the propaganda corpus fine tuning, and applying them on a corpus of political speeches in order to detect instances of propaganda.

III. Datasets

We leverage several datasets to train our model. The Proppy Corpus is our largest, featuring 52k news articles each labeled as "propagandistic" or "non-propagandistic" based on the news outlet it comes from. This can be taken as an indirect indication of the presence of a logical fallacy, as the associated paper [**?**] defines propaganda by the presence of any of seven

Logical Fallacy	Distribution	Example
Faulty Generalization	18.01%	I met a tall man who loved to eat cheese. Now I believe that all tall people like cheese.
Ad Hominem	12.33%	What can our new math teacher know? Have you seen how fat she is?
Ad Populum	9.47%	Everyone should like coffee: 95% of teachers do!
False Causality	8.82%	Every time I wash my car, it rains. Me washing my car has a definite effect on the weather
Circular Claim	6.98%	J.K. Rowling is a wonderful writer because she writes so well.
Appeal to Emotion	6.82%	It is an outrage that the school wants to remove the vending machines. This is taking our freedom away!
Fallacy of Relevance	6.61%	Why are you worried about poverty? Look how many children we abort every day.
Deductive Fallacy	6.21%	It is possible to fake the moon landing through special effects. Therefore, the moon landing was a fake using special effects.
Intentional Fallacy	5.84%	No one has ever been able to prove that extraterrestrials exist, so they must not be real.
Fallacy of Extension	5.76%	Their support of the discussion of sexual orientation issues is dangerous: they advocate for the exposure of children to sexually explicit materials, which is wrong.
False Dilemma	5.76%	You're either for the war or against the troops.
Fallacy of Credibility	5.39%	My professor, who has a Ph.D. in Astronomy, once told me that ghosts are real. Therefore, ghosts are real.
Equivocation	2.00%	I don't see how you can say you're an ethical person. It's so hard to get you to do anything; your work ethic is so bad.

TABLE I
Logical Fallacy Types in LogicEdu Dataset.

logical fallacies, including name calling and bandwagoning. Articles are labeled based on the trustworthiness of the news source each one originates from, as determined by the fact-checking website Media Bias/Fact Check. For example, the sentence "The Atlantic Coast Pipeline will bring more than natural gas to North Carolina" is labeled as not propagandistic, and the sentence "Grassroots leftists are planning an assault on the Supreme Court's judicial independence" is labeled as propagandistic. In total, 88% of the samples in this dataset are classified as non-propagandistic, while the rest are classified as propagandistic. This dataset is suitable for our task as it will provide a clear benchmark as to whether our model can appropriately identify propaganda.

Another dataset we are using is the LogicEdu dataset [**?**], which aggregates examples of logical fallacies found in student quiz websites and Google search results. See Table I for a breakdown of each label type. Each of the 2,217 samples in this dataset is a brief sentence, labeled with a specific logical fallacy. We use this dataset to train a language model that can differentiate between logical fallacies, with the ultimate goal of applying it towards the binary classification task via transfer learning.

Once our model is trained to a sufficient degree of performance on test data (using a portion of the dataset in [**?**]), we apply it towards a hand-picked selection of political speech. We selected six presidential speeches, two each from the administrations of George Washington, Donald Trump, and Joseph Biden. We treat each sentence in each speech as a separate sample, leading to a total of 772 samples. This dataset is not labeled as we do not intend to use it for training.

IV. Data Preprocessing

We preprocessed each of our datasets to include the same types of information, and with the intent to use as similar as possible a methodology to the fallacy detection papers we were referencing.

We first tokenized all of our texts in order to produce acceptable inputs for the language model. Since we use the BERT model, we use the corresponding tokenizer based on the WordPiece algorithm [**?**]. This is a subword-based algorithm, where unknown words are split into smaller meaningful subwords. We do not convert to lowercase to ensure that no valuable information is lost (the BERT model and the aforementioned WordPiece tokenizer algorithm are able to handle these properly).

In our later work, having frozen the layers of our BERT models we precomputed these values and saved them as part of pur preprocessing stage. This allowed us efficiency and speed improvements during the training and evaluation stages. To do this we first tokenized the documents, then computed and cached the model outputs at the last frozen layer. At train–time we only evaluated the upper layers of the model using as input the cached model outputs.

In summary, for each datapoint, our dataloaders produce a list of tokens corresponding to the text, or the cached BERT model outputs, depending on model type and a label corresponding to the fallacy type or propaganda label. We split both the LogicEdu and Proppy datasets into three components to use in various stages of the model training process: 70 percent for training, 15 percent for validation, and 15 percent for testing.

V. Methodology

Our general strategy is to perform transfer learning between the related tasks of Fallacy Classification and Propaganda Detection. We chose our model architecture based off of performant models described in Jin et. al. [**?**], noting that similar models performed well in the propaganda detection SemEval task [**?**]. In particular, we built our models off pretrained Large Language Models, aggregating across sentences and adding output layers. To ensure that our work yields a material improvement over simpler techniques, we also implemented a Naive Bayes classifier as a baseline.

Baseline Model:

We created a baseline model using a naive Bayes classifier. We preprocessed sentences from the text using the NLTK tokenizer and excluding all NLTK stopwords. Because of memory considerations all words occurring in less than 1%

of documents were omitted. We fit our model directly on the train fold of the Proppy dataset.

Initial Transfer Model:

We based our initial models off of those that performed best in the paper accompanying the LogicEdu dataset. This involved using a huggingface DistilBert model. DistilBert is a lighter and faster distilled version of BERT, first introduced in [**?**]. It is claimed to have 40% less parameters than bert-base-uncased, runs 60% faster while preserving over 95% of BERT's performance. It consists of an Embedding layer, followed by six transformer blocks, and a feed-forward network. To classify the sequence, our model additionally contained a linear layer converting to 768 outputs, and a final linear layer converting to the desired number of outputs (2 in the case of the Proppy dataset, and 13 for the LogicEdu dataset). We used cross entropy loss when training. All our training was done on Google Colab to take advantage of GPU usage. We experimented with freezing the weights on different sets of layers. In our final versions we froze the weights on the layers within the BERT model, and left the remaining uppermost layers unfrozen.

Cached Bert Transfer Models:

Our later models were designed to have a similar architecture to the first but use cached bert model outputs for efficiency purposes. This had a similar effect as freezing the layers of the BERT model. For these models we BERT rather than DistilBert, and precomputed the output values of the frozen layers of our model. The model consisted of BERT model outputs aggregated across the document followed by two output layers. We pretrained this model on the LogicEDU dataset for 200 epochs, then replaced the output layer and finetuned on the Proppy dataset for 800 epochs with a lower training rate.

We modified the loss function on the Propaganda task to account for the class imbalance in the dataset. Losses from the positive examples approximately 8 times more than than the respective negative examples. Doing so allowed us to achieve better performance on both regular and class-imbalance corrected test sets.

Additionally we compared this model with a second model which had not been pretrained on the LogicEDU dataset. This model had the same architecture as the previous one, except no pretraining was conducted.

Hyperparameter Selection:

For training, we use batch sizes of 16. We tuned the learning rate, dropout coefficents, model hidden sizes and weight decay values using a mix of handpicked values and gridsearch. We optimized for the accuracy metric on the unbalanced development subset of the Proppy dataset.

VI. Evaluation

We evaluated our candidate models on their performance classifying propagandistic texts, using the Proppy dataset and accuracy and F1 scores. For multiclass models, we evaluated performance using micro-averaged precision and recall across each logical fallacy type. We used the climate subest of the LogicEDU dataset for this purpose.

	DistillBert model	Cached-Bert model
Precision	0.142749	0.144807
Recall	0.162064	0.095000
F1	0.143275	0.114731
Accuracy	0.190000	0.095000

TABLE II
MULTICLASSIFICATION RESULTS USING BOTH MODEL TYPES ON TEST SET OF LOGIC EDU DATASET

VII. Results

Results on Baseline Model:

We created a baseline model using a naive Bayes classifier. We preprocessed sentences from the text using the NLTK tokenizer and excluding all NLTK stopwords. Because of memory considerations all words occurring in less than 1% of documents were omitted. We fit our classifier using add-α smoothing. In doing so we achieved 87% accuracy and an F-1 score of 57% on the test set.

Initial Transfer Models:

We also trained a BERT model on the LogicEdu dataset in preparation for transferring it to the Proppy binary classification task. Accuracy metrics per various hyperparameter settings are as follows:

Our results training DistilBERT on the LogicEdu dataset using the procedure described above in the Methodology section are shown in Table II. We tuned our hyperparameters to obtain optimum scores for our evaluation metrics, proceeding by keeping our learning rate relatively fixed while tweaking weight decay in increments of 0.1 between 1e-9 and 1e-5 and training on more epochs so that we can see convergence. Additionally, we attempted to reduce overfitting by placing other dropout layers in the model on 'train' mode, rather than the default 'eval' mode. However, while we were not able to increase the accuracy to a high threshold, we noted that this task also produced similar results to those in Jin et. al. and Barrón-Cedeño et. al. [**?**], [**?**]. We decided to proceed with transfer learning and assess our model's output in the next stage of training.

The results of our model's performance in the binary classification task on the Proppy Dataset is depicted in Table III. After tweaking hyperparameters in a similar range as when training on the LogicEdu dataset above, we were able to obtain satisfactory results across accuracy, precision, recall and F1 score, averaged around 77%. In order to ascertain if pretraining on logical fallacy data was truly improving the model's performance, we also trained the Proppy Dataset on a DistilBERT model that had not been pretrained on the LogicEdu data and compared the results. As shown in table III, pretraining the model on the LogicEdu dataset resulted in approximately 3% improvement across all metrics.

Cached Bert Transfer Models:

We trained the our cached bert transfer models in the same way as we did our regular ones. In particular, we found that modifying the loss function allowed us to correct for the

	Initial transfer model	Basic Cached-Bert model	Transferred Cached-Bert model
Precision	0.77327	0.963777	**0.972856**
Recall	0.772583	0.908173	**0.945781**
F1	0.770828	0.935149	**0.959127**
Accuracy	0.772595	0.905404	**0.928438**

TABLE III

RESULTS OF BOTH MODELS ON THE CLASS-IMBALANCED TEST SET OF THE PROPAGANDA DETECTION TASK

class imbalance in our training set, and learn a much more interesting rules. We did this on both the logical fallacy task, and the propaganda detection task. Interestingly doing so on the logical fallacy task produced worse results on it, but better results on the the fallacy detector transfer learned from it.

On the fallacy classification task, we tuned hyperparameters in increments of 0.1, first tuning model hidden size, then learning rate, then dropout probability then weight decay. We were able to achieve accuracies around 10%, and f1-scores of 12% on the test set with $lr = 1e-4$ a hidden size $h = 20$, dropout $p = 0.5$, and weight decay $d = 1e-5$. These metrics were slightly lower than those produced by Jin et. al. and Barrón-Cedeño et. al. [**?**], [**?**]; however, the model gave good results when considering the propaganda detection task.

In the propaganda detection task, we we tuned hyperparameters in the same way as we did for fallacy classification, with the exception of model achitecture, which we copied from the LogicEdu model to allow for transfer learning. We had our best performance on the development set with $lr = 1e-4$, dropout $p = 0.5$, and weight decay $d = 1e-8$. With that model we achieved accuracies of 87.5% and 92.8% and f-1 scores of 88.5% and 95.9% on the class imbalanced and balanced test sets, respectively.

For comparason we also trained a model without doing transfer learning. For this model, we tuned the hyperparameters from scrach again. Our best performant model achieved accuracies of 86.2% and 90.8% and f-1 scores of 86.9% and 93.5% on the class imbalanced and balanced test sets, respectively.

Results on Political Speeches Dataset:

To better understand what our model learned, we compiled the top most propagandistic and least propagandistic statements from our selection of presidential speeches, using the normalized positive and negative propaganda score outputs from our model, respectively. The results are shown below in tables V and VI. Evidently, the model has learned to classify extremist phrases, including those describing white supremacy, death, and violence, as more likely propagandistic, and fact-based statements like quotes and statistics as less likely propagandistic. Conversely, the model appears to be confused by speeches from George Washington, as they typically involved much longer sentences and more antiquated vocabulary rarely seen in the training datasets. Overall though, we notice a clear distinction between the two sets of classifications.

	Train	Test
Precision	0.459000	0.461192
Recall	0.691868	0.713043
F1	0.551875	0.560109
Accuracy	0.874451	0.874341

TABLE IV

BASELINE BINARY CLASSIFICATION RESULTS USING A NAIVE BAYES CLASSIFIER TO DETERMINE PROPAGANDISTIC CONTENT

VIII. DISCUSSION

In all observed metrics, the transfer learning model performed $2-5\%$ better than the non-transfer learning models on the propaganda detection task, even when holding the total number of training epochs constant. This is in part notable because our logical fallacy dataset was relatively small when compared with the propaganda detection dataset. We believe that this shows that the task of fallacy detection is significantly conceptually linked with the propaganda detection task, that there is meaningful overlap in the high level features important to each. This may be an interesting area of future exploration, especially with considering possible larger fallacy classification datasets, and other similar tasks are not uncommonly used.

Interestingly, although the cached-bert models performed slightly worse on the Logic Edu task, the resulting transfer learned models performed significantly better on the propaganda detection task. This might be for a number of reasons, but we advance a few possibilities. Firstly, the models used a smaller hidden size than those in the initial transfer model; so the fewer neurons which were transferred between the two models had greater information density and thus were more effective. Secondly, the Cached-bert model was trained using weighting to allow learning of features more independent of classes. This resulted in a slight drop in performance, but the resulting outputs had a larger variety of predicted classes indicating that the rule learned in the pre-training phase might itself have been more informative. Lastly, these models were able to be trained much faster due to the caching of the outputs of the frozen layers. As a result, significantly more training epochs and better hyperparameter tuning was possible for these models.

Additionally, we achieved similar performance in propaganda detection as Barrón-Cedeño et. al. [**?**], with only using text as an input to our models. This is significant, because Barrón-Cedeño et. al.'s propaganda detection system is dependent on external information in addition to the text of the article considered. Being able to achieve similar results using just the text as input, we believe marks a significant improvement in that it lowers overhead and can be applied in more flexible ways.

In the future, one of our main goals will be to increase the applications of the model on types of political speeches and the inferences we can draw from them. For instance, one goal

President	Statement	Pos. Rating
Joe Biden	*Not ISIS, not al Qaeda—white supremacists.*	0.7264887
Joe Biden	*This is a pandemic of the unvaccinated.*	0.7260819
Joe Biden	*Enough people who believed that America does not belong to everyone and not everyone is created equal—Native Americans, Asian Americans, Hispanic Americans, Black Americans.*	0.7222448
Joe Biden	*Literal hell was unleashed.*	0.715916
Joe Biden	*And most people didn't realize that, a century ago, a second Ku Klux Klan had been founded—the second Ku Klux Klan had been founded.*	0.7065899
Joe Biden	*This is totally unacceptable.*	0.70216185
George Washington	*No People can be bound to acknowledge and adore the invisible hand, which conducts the Affairs of men more than the People of the United States.*	0.6989093
Donald Trump	*Thank you, God bless you, and God bless America.*	0.69209516
Donald Trump	*Instead of saying "thank you" to the United States, they chanted "death to America." In fact, they chanted "death to America" the day the agreement was signed.*	0.6845047
Joe Biden	*Neo-Nazis, white supremacists, the KKK coming out of those fields at night in Virginia with lighted torches—the veins bulging on their—as they were screaming.*	0.6814902
Joe Biden	*As soon as they are authorized, those eligible will be able to get a booster right away in tens of thousands of site across the—sites across the country for most Americans, at your nearby drug store, and for free.*	0.68091244
Joe Biden	*These pandemic politics, as I refer to, are making people sick, causing unvaccinated people to die.*	0.67851
Joe Biden	*Well, Mother Fletcher said when she saw the insurrection at the Capitol on January the 9th [6th], it broke her heart—a mob of violent white extremists—thugs.*	0.6692622
Donald Trump	*You are attacking it, and you are attacking our country.*	0.66840583

TABLE V
Top Propagandistic Statements Retrieved From Political Speeches Dataset.

President	Statement	Neg. Rating
Joe Biden	*Friends gathered at music clubs and pool halls; at the Monroe family roller-skating rink.*	0.73096716
Joe Biden	*Talk about bullying in schools.*	0.73094827
Joe Biden	*The death toll records by local officials said there were 36 people.*	0.73018503
Joe Biden	*My plan will extend the vaccination requirements that I previously issued in the healthcare field.*	0.73009515
Joe Biden	*Mother Fletcher talks about how she was only able to attend school until the fourth grade and eventually found work in the shipyards, as a domestic worker.*	0.72985554
Joe Biden	*Words were exchanged.*	0.72954404
Donald Trump	*We will get through this challenge, just like we always do.*	0.7294126
Joe Biden	*Second, small businesses are the engines of our economy and the glue of our communities.*	0.72829604
Joe Biden	*The Russian government's borrowing rate spiked by over 15 percent.*	0.7282734
Joe Biden	*At the Dreamland Theatre, a young Black couple, holding hands, falling in love.*	0.7282581
Joe Biden	*There's nothing—not a single thing—we're unable to do if we do it together.*	0.7279522
Joe Biden	*But all John wanted to do was talk about how I was doing.*	0.7275841
Joe Biden	*All around, Black pride shared by the professional class and the working class who lived together, side by side, for blocks on end.*	0.72735226
Joe Biden	*It will hurt their ability to build ships, reducing their ability to compete economically.*	0.7271209
Donald Trump	*Tens of thousands of ISIS fighters have been killed or captured during my administration.*	0.72660106

TABLE VI
Top Non-Propagandistic Statements Retrieved From Political Speeches Dataset.

would be to evaluate whether there are any patterns in logical fallacies amongst politicians of different geographies, ideologies, or other defining characteristics. We could potentially accomplish this by using Speeches from the Congressional Record [**?**] as a top consideration, though we may also use an expanded dataset of presidential speeches.

We believe our language model's high performance in classifying propaganda is a positive indication in its future applications across a wide variety of real world problems concerning propaganda and misinformation detection. Furthermore, we found that performing transfer learning between logical fallacy based tasks and propaganda detection provided a significant improvement in model quality. Overall, these two improvements show promise in the field of text-based propaganda detection, and we believe that the tools we present in this paper can be useful along that avenue.

IX. Code

All code for this project can be found on our git repository at https://github.com/ZachEichen/propaganda_detection.