PROTECT
A Pipeline for Propaganda Detection and Classification

Vorakit Vorakitphan, Elena Cabrio, Serena Villata
Université Côte d'Azur, Inria, CNRS, I3S, France
vorakit.vorakitphan@inria.fr,
elena.cabrio@univ-cotedazur.fr, villata@i3s.unice.fr

Abstract

English. Propaganda is a rhetorical technique to present opinions with the deliberate goal of influencing the opinions and the actions of other (groups of) individuals for predetermined misleading ends. The employment of such manipulation techniques in politics and news articles, as well as its subsequent spread on social networks, may lead to threatening consequences for the society and its more vulnerable members. In this paper, we present PROTECT (PROpaganda Text dEteCTion), a new system to automatically detect propagandist messages and classify them along with the propaganda techniques employed. PROTECT is designed as a full pipeline to firstly detect propaganda text snippets from the input text, and then classify the technique of propaganda, taking advantage of semantic and argumentation features. A video demo of the PROTECT system is also provided to show its main functionalities.

Italiano. *La propaganda è una tecnica retorica per presentare determinate opinioni con l'obiettivo deliberato di influenzare le opinioni e le azioni di altri (gruppi di) individui per fini predeterminati e tendenzialmente fuorvianti. L'impiego di tale tecnica di manipolazione in politica e nella stampa, così come la sua diffusione sulle reti sociali, può portare a conseguenze disastrose per la società e per i suoi membri più vulnerabili. In questo articolo presentiamo PROTECT (PROpaganda Text dEteCTion), un nuovo sistema per identificare automaticamente i messaggi propagandistici e classificarli rispetto alle tecniche di propaganda utilizzate. PROTECT è un sistema progettato come una pipeline completa per rilevare in primo luogo i frammenti di testo propagandistici dato il testo proposto, e successivamente classificare tali frammenti secondo la tecnica di propaganda usata, sfruttando le caratteristiche semantiche e argomentative del testo. Questo articolo presenta anche un video dimostrativo del sistema PROTECT per mostrare le principali funzionalità fornite all'utente.*

1 Introduction

Propaganda represents an effective but often misleading communication strategy which is employed to promote a certain viewpoint, for instance in the political context (Lasswell, 1938; Koppang, 2009; Dillard and Pfau, 2009; Longpre et al., 2019). The goal of this communication strategy is to persuade the audience about the goodness of such a viewpoint by means of misleading and/or partial arguments, which is particularly harmful for the more vulnerable public in the society (e.g., young or elder people). Therefore the ability to detect the occurrences of propaganda in political discourse and newspaper articles is of main importance, and Natural Language Processing methods and technologies play a main role in this context addressing the propaganda detection and classification task (Da San Martino et al., 2019; Da San Martino et al., 2020a). It is, in particular, important to make this vulnerable public aware of the problem and provide them tools able to raise their awareness and develop their critical thinking.

To achieve this ambitious goal, we present in

this paper a new tool called PROTECT (PROpaganda Text dEteCTion) to automatically identify and classify propaganda in texts. In the current version, only English text is processed. This tool has been designed with an easy-to-access user interface and a web-service API to ensure a wide public use of PROTECT online. To the best of our knowledge, PROTECT is the first online tool for propagandist text identification and classification with an interface allowing the user to submit his/her own text to be analysed.[1]

PROTECT presents two main functionalities: *i)* the automatic propaganda detection and classification service, which allows the user to paste or upload a text and returns the text where the propagandist text snippets are highlighted in different colors depending on the propaganda technique which is employed, and *ii)* the propaganda word clouds, to show in a easy to catch visualisation the identified propagandist text snippets. PROTECT is deployed as a web-service API, allowing users to download the output (the text annotated with the identified propaganda technique) as a `json` file. The PROTECT tool relies on a pipeline architecture to first detect the propaganda text snippets, and second to classify the propaganda text snippets with respect to a specific propaganda technique. We cast this task as a sentence-span classification problem and we address it relying on a transformer architecture. Results reach SoTA systems performances on the tasks of propaganda detection and classification (for a comparison with SoTA algorithms, we refer to (Vorakitphan et al., 2021)).

The paper is structured as follows: first, Section 2 discusses the state of the art in propaganda detection and classification and compares our contribution to the literature. Then Section 3 describes the pipeline for the detection and classification of propaganda text snippets as well as the data sets used for the evaluation and the obtained results. Section 4 describes the functionalities of the web interface, followed by the Conclusions.

2 Related Work

In the last years, there has been an increasing interest in investigating methods for textual propaganda detection and classification. Among them, (Barrón-Cedeño et al., 2019) present a system to organize news events according to the level of propagandist content in the articles, and introduces a new corpus (QProp) annotated with the propaganda vs. trustworthy classes, providing information about the source of the news articles. Recently, a web demo named `Prta` (Da San Martino et al., 2020b) has been proposed, trained on disinformation articles. This demo allows a user to enter a plain text or a URL, but it does not allow users to download such results. Similarly to PROTECT, `Prta` shows the propagandist messages at the snippet level with an option to filter the propaganda techniques to be shown based on the confidence rate, and also analyzes the usage of propaganda technique on determined topics. The implementation of this system relies on the approach proposed in (Da San Martino et al., 2019).

The most recent approaches for propaganda detection are based on language models that mostly involve transformer-based architectures. The approach that performed best on the NLP4IF'19 sentence-level classification task relies on the BERT architecture with hyperparameters tuning without activation function (Mapes et al., 2019). (Yoosuf and Yang, 2019) focused first on the pre-processing steps to provide more information regarding the language model along with existing propaganda techniques, then they employ the BERT architecture casting the task as a sequence labeling problem. The systems that took part in the SemEval 2020 Challenge - Task 11 represent the most recent approaches to identify propaganda techniques based on given propagandist spans. The most interesting and successful approach (Jurkiewicz et al., 2020) proposes first to extend the training data from a free text corpus as a silver dataset, and second, an ensemble model that exploits both the gold and silver datasets during the training steps to achieve the highest scores.

As most of the above mentioned systems, also PROTECT relies on language model architectures for the detection and classification of propaganda messages, empowering them with a rich set of features we identified as pivotal in propagandist text from computational social science literature (Vorakitphan et al., 2021). In particular, (Morris, 2012) discusses how emotional markers and affect at word- or phrase-level are employed in propaganda text, whilst (Ahmad et al., 2019) show that the most effective technique to extract senti-

[1] The video demonstrating the PROTECT tool is available here `https://1drv.ms/u/s!Ao-qMrhQAfYtkzD69JPAYY3nSFub?e=oUQbxQ`

ment for the propaganda detection task is to rely on lexicon-based tailored dictionaries. (Li et al., 2017) show how to detect degrees of strength from calmness to exaggeration in press releases. Finally, (Troiano et al., 2018) focus on feature extraction of text exaggeration and show that main factors include imageability, unexpectedness, and the polarity of a sentence.

3 Propaganda Detection and Classification

PROTECT addresses the task of propaganda technique detection and classification at fragment-level, meaning that both the spans and the type of propaganda technique are identified and highlighted in the input sentences. In the following, we describe the datasets used to train and test PROTECT, and the approach implemented in the system to address the task.

3.1 Datasets

To evaluate the approach on which PROTECT relies, we use two standard benchmarks for Propaganda Detection and Classification, namely the NLP4IF'19 (Da San Martino et al., 2019) and SemEval'20 datasets (Da San Martino et al., 2020a). The former was made available for the shared task NLP4IF'19 on fine-grained propaganda detection. 18 propaganda techniques are annotated on 469 articles (293 in the training set, 75 in the development set, and 101 in the test set).[2] As a follow up, in 2020 SemEval proposed a shared task (T11)[3] reducing the number of propaganda categories with respect to NLP4IF'19 (14 categories, 371 articles in the training set and 75 in the development set). PROTECT detects and classifies the same list of 14 propaganda techniques as in the SemEval task, namely: *Appeal_to_Authority, Appeal_to_fear-prejudice, Bandwagon, Reductio_ad_hitlerum, Black-and-White_Fallacy, Causal_Oversimplification, Doubt, Exaggeration_Minimisation, Flag-Waving, Loaded-Language, Name-Calling_Labeling, Repetition, Slogans, Thought-terminating_Cliches, Whataboutism_Straw-Men_Red-Herring*.

Those classes are not uniformly distributed in the data sets. *Loaded-Language* and *Name-Calling_Labeling* are the classes with the higher number of instances (representing respectively 32% and 15% of the propagandist messages on all above-mentioned datasets). The classes with the lower number of instances are *Whataboutism, Red-Herring, Bandwagon, Straw-Men*, respectively occurring in 1%, 0.87%, 0.29%, 0.23% in NLP4IF'19 datasets. In SemEval'20T11 such labels where merged, and the classes *Whataboutism_Straw-Men_Red-Herring, Bandwagon* respectively represent 1.33% and 1.29% of the propagandist messages.

3.2 PROTECT Architecture

Given a textual document or a paragraph as input, the system performs two steps. First, it performs a binary classification at token level, to label a token as propagandist or not. Then, it classifies propagandist tokens according to the 14 propaganda categories from SemEval task (T11).

For instance, given the following example *"Manchin says Democrats acted like babies at the SOTU (video) Personal Liberty Poll Exercise your right to vote."* the snippets *"babies"* is first classified as propaganda (step 1), and then more specifically as an instance of the *Name-Calling_Labeling* propaganda technique (step 2).

Step 1: Propaganda Snippet Detection. To train PROTECT, we merge the training, development and test sets from NLP4IF, and the training set from Semeval'20 T11. The development set from Semeval'20 T11 is instead used to evaluate the system performances.[4] In the preprocessing phase, each sentence is tokenized and tagged with a label per token according to the IOB format.

For the binary classification, we adopt *Pretrained Language Model* (PLM) based on BERT (*bert-base-uncased* model) (Devlin et al., 2019) architecture. The hyperparameters are a learning rate of 5e-5, a batch of 8, max_len of 128. For the evaluation, we compute standard classification metrics[5] at the token-level. The results obtained by the binary classifier (macro average over 5 runs) on SemEval'20 T11 development set are 0.71 precision, 0.77 recall and 0.72 F-measure (us-

[2] https://propaganda.qcri.org/nlp4if-shared-task/

[3] https://propaganda.qcri.org/semeval2020-task11/

[4] The gold annotations of Semeval'20 test set are not available, this is why we selected the development set for evaluation.

[5] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support.html

Propaganda Technique	PLM: RoBERTa
Appeal_to_Authority	0.48
Appeal_to_fear-prejudice	0.57
Bandwagon,Reductio_ad_hit.	0.72
Black-White-Fallacy	0.38
Casual-Oversimplification	0.70
Doubt	0.74
Exaggeration,Minimisation	0.67
Flag-Waving	0.88
Loaded_Language	0.88
Name_Calling,Labeling	0.85
Repetition	0.70
Slogans	0.72
Thought-terminating_Cliches	0.52
Whatab.,Straw_Men,Red_Her.	0.55
Average	0.67

Table 1: Results on sentence-span classification on SemEval'20 T11 dev set (micro-F1) using span-pattern produced by the binary classification step (Step 1).

ing Softmax as activation function[6]).

We then perform a post-processing step to automatically join tokens labelled with the same propaganda technique into the same textual span.

Given that PLM is applied at token-level, each token is processed into sub-words (e.g., "running" is tokenized and cut into two tokens: "run" and "##ing"). Such sub-words can mislead the classifier. For instance, in the following sentence: "The next day, Biden said, he was informed by Indian press that there were at *least a few Bidens in* India.", our system detects *least a few Bidens in* as a propagandist snippet, but it misclassifies one sub-word ("at" was not considered as part of "at least", and therefore excluded from the propagandist snippet).

Step 2: Propaganda Technique Classification. We cast this task as a sentence-span multi-class classification problem. More specifically, both the tokenized sentence and the span are used to feed the transformer-based model RoBERTa (*roberta-base* pre-trained model)[7] (Liu et al., 2019) to perform both a sentence classification and a span classification. More precisely: *i)* we input a sentence to the tokenizer where max_length is set to 128 with padding; *ii)* we input the span provided by the propaganda span-template from SemEval T11 dataset, and we set max_length value of 20 with padding. RoBERTa tokenizer is applied in both cases. If a sentence does not contain propaganda spans, it is labeled as "none-propaganda".

To take into account context features at sentence-level, a BiLSTM is introduced. For each sentence, semantic and argumentation features are extracted following the methodology proposed in (Vorakitphan et al., 2021) and given in input to the BiLSTM model (hyper-parameters: 256 hidden_size, 1 hidden_layer, drop_out of 0.1 with ReLU function at the last layer before the joint loss function). Such features proved to be useful to improve the performances of our approach on propagandist messages classification, obtaining SoTA results on some categories (in (Vorakitphan et al., 2021) we provide a comparison of our model with SoTA systems on both NLP4IF and SemEval datasets).

To combine the results from sentence-span based RoBERTa with the feature-based BiLSTM we apply the joint loss strategy proposed in (Vorakitphan et al., 2021). Each model produces a loss per batch using CrossEntropy loss function L. Following the function: $loss_{\text{joint\_loss}} = \alpha \times \frac{(loss_{\text{sentence}} + loss_{\text{span}} + loss_{\text{semantic\_argumentation\_features}})}{N_{\text{loss}}}$ where each $loss$ value is produced from CrossEntropy function of its classifier (e.g., $loss_{\text{sentence}}$ and $loss_{\text{span}}$ from RoBERTa models of sentence and span, $loss_{\text{semantic\_argumentation\_features}}$ from the BiLSTM model.)

To train the above mentioned methods for the propaganda technique classification task, we merged the data sets of NLP4IF'19 and SemEval'20 T11 (same setting as in Step 1). Then we tested the full pipeline of PROTECT on the development set from Semeval'20 T11. The output of the snippet detection task (Step 1) are provided as a span-pattern to the models performing Step 2. Table 1 reports on the obtained results of the full pipeline (Step 1+Step 2) averaged over 5 runs (we cannot provide a fair comparison of those results with SoTA systems, given that in SemEval the two tasks are separately evaluated and no pipeline results are provided). We can notice however, that our results in a pipeline are comparable with the

[6]We are aware that sigmoid function is usually used as default activation function in binary classification. However, in our setting we tested both functions and we obtained better performances with Softmax as activation function (+0.04 F1 with respect to sigmoid).

[7]https://huggingface.co/transformers/model_doc/roberta.html

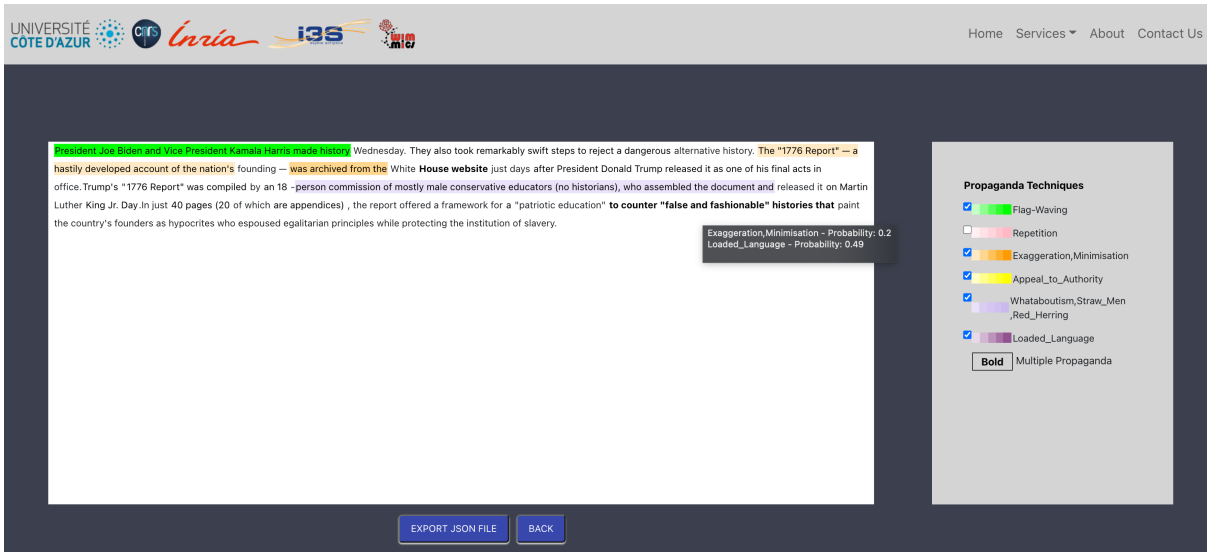

Figure 1: PROTECT Interface: Propaganda Techniques Classification

ones obtained in (Vorakitphan et al., 2021) on the two separate tasks.

Given the high complexity of the propaganda technique classification task and the classes' unbalance, some examples are miss-classified by the system. For instance, in the following sentence "The Mueller probe saw several within Trump's orbit indicted, but not *Trump , as family* or Trump himself", the system annotated the snippet in italics as "Name_Calling,Labeling", while the correct labels would have been "Repetition".

4 PROTECT Functionalities

As previously introduced, PROTECT allows a user to input plain text and retrieve the propagandist spans in the message as output by the system. In the current version of the system, two services are provided through the web interface (and the API), described in the following.

4.1 Service 1: Propaganda Techniques Classification

The system accepts an input plain text in English, and then the architecture described in Section 3.2 is run over such text. The output consists of an annotated version of the input text, where the different propagandist techniques detected by the system are highlighted in different colours. The colour of the highlighted snippet is distinctive of a certain propaganda technique: the darker the color, the higher the confidence score of the system in assigning the label to a textual snippet. Figure 1 shows an example of PROTECT web interface. Checkboxes on the right side of the page provide the key to interpret the colors, and allow the user to check or un-check (i.e. highlight or not) the different propagandist snippets in the text, filtering the results. Faded to dark colours represent the confidence level of the prediction (the darker the colour, the higher the system confidence). The snippets in bold contain multiple propaganda techniques in the same text spans, that can be unveiled hovering with the mouse over the snippets.

As said before, PROTECT can be used through the provided API, and annotated text can be downloaded as a JSON file with the detected propagandist snippet(s) at character indices (start to end indices of a snippet) based on individual sentence, propaganda technique(s) used, and the confidence score(s)).

4.2 Service 2: Propaganda Word Clouds

The propagandist snippets output by the system can also be displayed as word clouds, where the size of the words represents the system confidence score in assigning the labels (see Figure 2). The different sizes represent the confidence score of the prediction, and the colors the propaganda technique (as in Service 1). If multiple techniques are found in the same snippet, it is duplicated in the word cloud. As for the first service, a checkbox on the right side of the word clouds allows the user to select the propagandist techniques to be visualized. Also in this case, a json file can be downloaded with the system prediction.

The word cloud service has been added to PRO-

Figure 2: PROTECT Interface: Word Cloud

TECT in addition to the standard visualization, to provide a different and informative way to summarise propaganda techniques on a topic, and to facilitate their identification.

5 Conclusions

In this paper, we presented PROTECT, a propaganda detection and classification tool. PROTECT relies on a pipeline to detect propaganda snippets from plain text. We evaluated the proposed pipeline on standard benchmarks achieving state-of-the-art results. PROTECT is deployed as a web-service API that accepts a plain text input, returning downloadable annotated text for further usage. In addition, a propaganda word clouds service allows to gain further insights from such text.

Acknowledgments

This work is partially supported by the ANSWER project PIA FSN2 n. P159564-2661789/DOS0060094 between Inria and Qwant. This work has also been supported by the French government, through the 3IA Côte d'Azur Investments in the Future project managed by the National Research Agency (ANR) with the reference number ANR-19-P3IA-0002.

References

Siti Rohaidah Ahmad, Muhammad Zakwan Muhammad Rodzi, Nurlaila Syafira Shapiei, Nurhafizah Moziyana Mohd Yusop, and Suhaila Ismail. 2019. A review of feature selection and sentiment analysis technique in issues of propaganda. *International Journal of Advanced Computer Science and Applications*, 10(11).

Alberto Barrón-Cedeño, Israa Jaradat, Giovanni Martino, and Preslav Nakov. 2019. Proppy: Organizing the news based on their propagandistic content. *Information Processing & Management*, 56, 05.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019. Fine-grained analysis of propaganda in news article. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5636–5646, Hong Kong, China, November. Association for Computational Linguistics.

Giovanni Da San Martino, Alberto Barrón-Cedeño, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020a. SemEval-2020 task 11: Detection of propaganda techniques in news articles. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1377–1414, Barcelona (online), December. International Committee for Computational Linguistics.

Giovanni Da San Martino, Shaden Shaar, Yifan Zhang, Seunghak Yu, Alberto Barrón-Cedeño, and Preslav Nakov. 2020b. Prta: A system to support the analysis of propaganda techniques in the news. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 287–293, Online, July. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

James Price Dillard and Michael Pfau. 2009. *The Persuasion Handbook: Developments in Theory and Practice*. Sage Publications, Inc.

Dawid Jurkiewicz, Łukasz Borchmann, Izabela Kosmala, and Filip Graliński. 2020. ApplicaAI at SemEval-2020 task 11: On RoBERTa-CRF, span CLS and whether self-training helps them. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1415–1424, Barcelona (online), December. International Committee for Computational Linguistics.

Haavard Koppang. 2009. Social influence by manipulation: A definition and case of propaganda. *Middle East Critique*, 18:117 – 143.

Harold Dwight Lasswell. 1938. Propaganda technique in the world war.

Yingya Li, Jieke Zhang, and Bei Yu. 2017. An NLP analysis of exaggerated claims in science news. In *Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism*, pages 106–111, Copenhagen, Denmark, September. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692. _eprint: 1907.11692.

Liane Longpre, Esin Durmus, and Claire Cardie. 2019. Persuasion of the undecided: Language vs. the listener. In *Proceedings of the 6th Workshop on Argument Mining*, pages 167–176, Florence, Italy, August. Association for Computational Linguistics.

Norman Mapes, Anna White, Radhika Medury, and Sumeet Dua. 2019. Divisive language and propaganda detection using multi-head attention transformers with deep learning BERT-based language models for binary classification. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 103–106, Hong Kong, China, November. Association for Computational Linguistics.

Travis Morris. 2012. Extracting and networking emotions in extremist propaganda. In *2012 European Intelligence and Security Informatics Conference*, pages 53–59.

Enrica Troiano, Carlo Strapparava, Gözde Özbal, and Serra Sinem Tekiroğlu. 2018. A computational exploration of exaggeration. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3296–3304, Brussels, Belgium, October-November. Association for Computational Linguistics.

Vorakit Vorakitphan, Elena Cabrio, and Serena Villata. 2021. "Don't discuss": Investigating Semantic and Argumentative Features for Supervised Propagandist Message Detection and Classification. In *Recent Advances in Natural Language Processing (RANLP 2021)*, Varna (Online), Bulgaria, September.

Shehel Yoosuf and Yin Yang. 2019. Fine-grained propaganda detection with fine-tuned BERT. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 87–91, Hong Kong, China, November. Association for Computational Linguistics.