Hierarchical Multi-Instance Multi-Label Learning for Detecting Propaganda Techniques

Anni Chen and **Bhuwan Dhingra**
Department of Computer Science
Duke University
Durham, NC, USA
anni.chen@alumni.duke.edu
bhuwan.dhingra@duke.edu

Abstract

Since the introduction of the SemEval 2020 Task 11 (Martino et al., 2020a), several approaches have been proposed in the literature for classifying propaganda based on the rhetorical techniques used to influence readers. These methods, however, classify one span at a time, ignoring dependencies from the labels of other spans within the same context. In this paper, we approach propaganda technique classification as a Multi-Instance Multi-Label (MIML) learning problem (Zhou et al., 2012) and propose a simple RoBERTa-based model (Zhuang et al., 2021) for classifying all spans in an article simultaneously. Further, we note that, due to the annotation process where annotators classified the spans by following a decision tree, there is an inherent hierarchical relationship among the different techniques, which existing approaches ignore. We incorporate these hierarchical label dependencies by adding an auxiliary classifier for each node in the decision tree to the training objective and ensembling the predictions from the original and auxiliary classifiers at test time. Overall, our model leads to an absolute improvement of 2.47% micro-F1 over the model from the shared task winning team in a cross-validation setup and is the best performing non-ensemble model on the shared task leaderboard.

1 Introduction

The development of the Web and social media has amplified the scale and effectiveness of propaganda (Barrón-Cedeno et al., 2019). Automatic propaganda detection through text analysis enables social science researchers to analyze its spread at scale (Glowacki et al., 2018; Martino et al., 2020b). Within text analysis, two broad approaches include the identification of rhetorical techniques used to influence readers and document-level propagandistic article classification. The former is more promising because propaganda techniques are easy to identify and are the very building blocks (Martino et al., 2020b). Hence, it is the focus of this paper.

Recent research on fine-grained propaganda detection has been spurred by the NLP4IF-2019 Shared Task (Da San Martino et al., 2019) and its follow-up SemEval 2020 Task 11 (Martino et al., 2020a). In this paper, we focus on its Technique Classification subtask where, given a news article text and spans identified as propagandistic, systems need to classify each of the spans into one or more of 14 different common propaganda techniques.

All top systems submitted to the task (Jurkiewicz et al., 2020; Chernyavskiy et al., 2020; Morio et al., 2020; Raj et al., 2020) employed a pretrained RoBERTa-based model (Zhuang et al., 2021) which was trained to classify one span at a time. However, labels of different spans within the same article clearly depend on each other. Thus, we approach the task within a Multi-Instance Multi-Label framework (Zhou et al., 2012), where we model each article as an object with multiple instances (spans), each with its own labels. This allows us to model the dependencies between different labels within the same article. We show that this $\text{MIML}_{\text{RoBERTa}}$ observes a 1.98% micro-F1 improvement over the replicated ApplicaAI system (referred to as baseline) (Jurkiewicz et al., 2020).

Besides, as a decision tree was used to guide annotations (Martino et al., 2020a), we explore incorporating this hierarchical relationship among the labels into classifiers. To do so, we add 7 more auxiliary classifiers on top of the span representations from RoBERTa, one for each intermediate node in the tree, and train these classifiers to predict the path to a leaf node and hence the corresponding label (see Figure 1). We show that incorporating the label hierarchy in this manner improves both the single-instance and the MIML versions of the RoBERTa model (referred to as hierarchical baseline and hierarchical $\text{MIML}_{\text{RoBERTa}}$ respectively).

Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023), pages 155–163
July 13, 2023 ©2023 Association for Computational Linguistics

2 Related Work

The MIML framework was first introduced by Zhou et al. (2012) aiming for better representing complicated objects composed of multiple instances, e.g., an image with several bounding boxes each with its own label. Since then, it has been applied to many tasks, such as relation extraction (Surdeanu et al., 2012) and aspect-category sentiment analysis (Li et al., 2020). The latter work uses a Bi-LSTM architecture which aggregates over the words in a sentence (instances) to classify the sentiments of different aspects (labels). MIML has also been applied to BERT-based models in the biomedical text analysis (Tian and Zhang, 2021). To our best knowledge, our work is the first one to apply it for propaganda technique classification.

Given the decision tree used for annotations (reproduced in Figure 4 in Appendix), this task can also be viewed as a hierarchical text classification problem, with mandatory leaf node prediction and a tree-structured hierarchy (Silla and Freitas, 2011). Exploiting hierarchical information has been useful in significantly enhancing the performance of the system for medical image annotations (Dimitrovski et al., 2011) and presents us with an opportunity to apply to the propaganda method detection. Similar to (Dumais and Chen, 2000; Weigend et al., 1999), we use the multiplicative rule to combine probabilities along the different paths in the hierarchy, leading to a distribution across the leaf nodes which can be combined with the distribution predicted by the non-hierarchical baseline. Different approaches to hierarchical classification across multiple domains can be found in Silla and Freitas (2011).

3 Methods[1]

Task Description. Given an article d with propagandistic spans $s_1, ..., s_m$ identified by their start and end indices, the task is to identify the sets of techniques $y_1, ..., y_m$ where $y_i \subseteq Y \; \forall i \in \{1, ..., m\}$ and Y represents the set of 14 techniques. Following the shared task, we assume that the number of labels $|y_i|$ for each span is known.

3.1 Single-Instance Baseline

The baseline system from the winning ApplicaAI team (Jurkiewicz et al., 2020) uses a RoBERTa-based classifier and applies to each span separately. The span is padded by special tokens <bop> and <eop> on either side. A total context of 256 tokens on both sides of the target span spanning multiple sentences are included in the input, unlike other systems (Chernyavskiy et al., 2020; Morio et al., 2020; Raj et al., 2020) on the leaderboard which limit to the target sentence. For classification, we use the <bop> representation output from RoBERTa and pass it through a linear layer followed by softmax. Jurkiewicz et al. (2020) further show marginal improvements by re-weighting the loss for under-represented classes, self-training over unlabeled data and a Span-CLS approach which adds a second transformer network on top of RoBERTa only over the tokens in the target span. In our experiments, we did not use any of the improvements.

3.2 MIML$_{\text{RoBERTa}}$

During the initial inspection of data, we observed high absolute pointwise mutual information between certain techniques. For example, in an article, if the *slogans* technique appears, the *flag-waving* technique usually follows. This observation motivates us to predict the labels of *all* spans in a text simultaneously.

During preprocessing, we pad the spans with pairs of <bop0>, <eop0>, <bop1>, <eop1>... to indicate the start and the end of a text fragment, even in cases where some spans are nested inside or overlapping with others. After padding, every article is split into windows of size 512 tokens with a stride of 256 tokens until there are no more labeled spans. Whenever there are spans that need to be truncated as we create a window, its corresponding <eop> is appended near the end of the window to ensure that the number of <bop>s and <eop>s match. In the case where there is a specific nesting structure between any two text fragments, its nesting structure is respected in appending the <eop> to represent the differences in text fragments even after truncation. Also, since only around 1.8% of the total annotations are spans with multiple labels (Martino et al., 2020a), for ease of implementation, we use the rarer label for every text span during preprocessing and the later training.

Let h_s denote the RoBERTa output at the <bop> for a span s, then we compute the logits for each of the labels as $p_{\text{flat}}(c) \propto \exp(h_s \cdot w_c)$, for $c = 0, \ldots, 13$. The model is trained using the cross-

[1] Code is available at https://github.com/Dranoxgithub/propaganda-nlp-new.

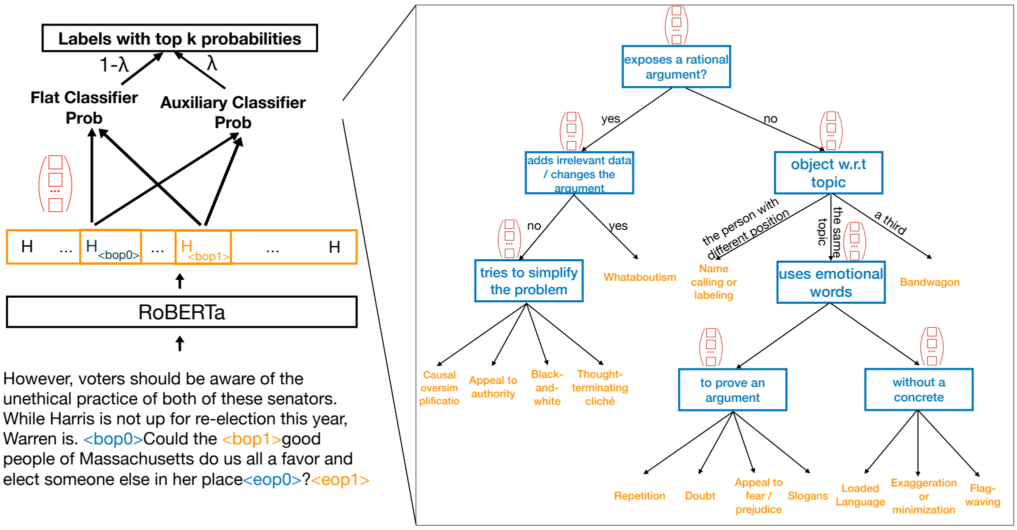
Labels with top k probabilities
1-λ
λ
Flat Classifier Prob
Auxiliary Classifier Prob
H
...
H<bop0>
...
H<bop1>
...
H
RoBERTa
However, voters should be aware of the unethical practice of both of these senators. While Harris is not up for re-election this year, Warren is. <bop0>Could the <bop1>good people of Massachusetts do us all a favor and elect someone else in her place<eop0>?<eop1>
exposes a rational argument?
yes
no
adds irrelevant data / changes the argument
object w.r.t topic
no
yes
tries to simplify the problem
Whataboutism
the person with different position
the same topic
a third
Name calling or labeling
uses emotional words
Bandwagon
Causal oversim plificatio
Appeal to authority
Black-and-white
Thought-terminating cliché
to prove an argument
without a concrete
Repetition
Doubt
Appeal to fear / prejudice
Slogans
Loaded Language
Exaggeration or minimization
Flag-waving

Figure 1: Overview of the hier $\text{MIML}_{\text{RoBERTa}}$. Multiple spans within the same text (instances) are classified together. We add an auxiliary classifier to the model at each intermediate node. Decision tree used for annotations (right).

entropy loss:

$$l_{\text{flat}} = \sum_{c=1}^{C} -\mathbb{1}_{y=c} \log p_{\text{flat}}(c), \qquad (1)$$

where w_c is a weight vector and y is the ground truth label of the span. We compute l_{flat} simultaneously for all spans in the text and take an average.

During inference, we predict the number of labels requested for each span by selecting the classes with the highest in p_{flat}. Since some spans can appear in multiple windows, the set of predictions from the window where the span is least truncated and has the most surrounding context is chosen.

3.3 Hierarchical $\text{MIML}_{\text{RoBERTa}}$

Due to the complex and subjective nature of the task, span annotations by Martino et al. (2020a) were guided by a decision tree. For example, annotators first consider whether the argument is rational, followed by whether emotional words are used, and so on. This process induces a hierarchy among the labels, and to model such relationships, we use a simplified version of the decision tree whose leaf nodes are the 14 propaganda techniques (Figure 1, right). We add an auxiliary loss to the training objective based on hierarchical text classification which trains local classifiers at each intermediate node (Silla and Freitas, 2011).

Let K be the number of intermediate nodes and hence the number of auxiliary classifiers, and let $C_1, C_2, \ldots, C_K$ denote the number of outgoing edges or the number of labels for each classifier. Then, given the RoBERTa representation for a span h_s, we compute the probability of following an edge i from an intermediate node k as:

$$p_k(i) \propto \exp(h_s \cdot w_{k,i}), \quad \forall i = 0, \ldots, C_k \qquad (2)$$

where $w_{k,i}$ is a weight vector and the probabilities are normalized across the C_k labels for each classifier. Given a leaf node label c, we denote the path of classifiers and edges from the root to it as $I_c = \{(k_1, i_1), (k_2, i_2), \ldots\}$, where each tuple in the set denotes a pair of classifier and its corresponding label along the path. Then we can compute the overall probability of selecting as:

$$p_{\text{aux}}(c) = \prod_{(k,i) \in I_c} p_k(i). \qquad (3)$$

Note that p_{aux} forms a valid distribution over the leaf nodes since each probability along the path is normalized.

During training, we compute an auxiliary loss for each span which minimizes the negative log-likelihood of selecting the edges along the path to its correct label y:

$$l_{\text{aux}} = \frac{1}{|I_y|} \sum_{(k,i) \in I_y} \sum_{c=1}^{C_k} -\mathbb{1}_{i=c} \log p_k(i). \qquad (4)$$

$\lambda_{\text{training}}$ and λ_{eval} are both hyperparameters. The overall training loss is a combination of the flat loss described in the previous section and the auxiliary loss above: $l_{\text{ovr}} = (1 - \lambda_{\text{training}}) \cdot l_{\text{flat}} + \lambda_{\text{training}} \cdot l_{\text{aux}}$.

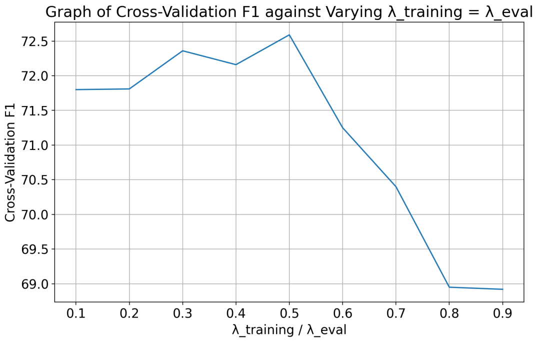

Figure 2: Cross-validation F1 against varying lambda values where $\lambda_{\text{training}} = \lambda_{\text{eval}}$. The performance is the best at $\lambda_{\text{training}} = \lambda_{\text{eval}} = 0.5$.

During inference, we combine the predicted probabilities over the labels from both the flat classifier and auxiliary classifiers: $p_{\text{ovr}}(c) = (1 - \lambda_{\text{eval}}) \cdot p_{\text{flat}}(c) + \lambda_{\text{eval}} \cdot p_{\text{aux}}(c)$.

4 Experiments

4.1 Dataset

The original training and development set respectively contains 357, 74 articles, and thus correspondingly 6128, 1063 data points. Following Jurkiewicz et al. (2020), we also evaluate using six-fold cross-validation where the folds are created by mixing the original training and development sets. As a result each fold roughly consists of 6000 training and 1200 evaluation data points. Below we report the average and standard deviation of the metrics across six folds. We also submit our best models, trained on the original training set, to the leaderboard for evaluation on the official test set.

4.2 Training & Evaluation

We tune both $\lambda_{\text{training}}$ between 0 and 1 with a step of 0.1. For every value of $\lambda_{\text{training}}$, we also tune λ_{eval} at 0, 1 and $\lambda_{\text{training}}$.The model's performance peaks at $\lambda_{\text{training}} = \lambda_{\text{eval}} = 0.5$ (Figure 2) and all the results reported use this set of values. Hyperparameter details are included in Appendix B.

The scorer script provided by the shared task organizers evaluates a micro-averaged F1 score, taking the best match between the predictions and ground truth labels when a span has multiple labels (Martino et al., 2020a). We use this script for evaluation in all our experiments. Additionally, we also use a *Tree-F1* score from the hierarchical text classification literature which measures the overlap between the paths from the root node to the ground truth and predicted nodes (Kosmopoulos et al., 2015) (details in Appendix C). In particular, we are interested in seeing if models with hierarchical information make mistakes closer to the ground truth in the tree.

5 Results and Discussion

Method		Cross validation F1(%)
Baseline	non-hier	70.12 ± 2.06
	hier	70.49 ± 2.01
MIML$_{\text{RoBERTa}}$	non-hier	72.10 ± 0.83
	random	69.54 ± 1.33
	hier	$\mathbf{72.59 \pm 1.02}$

Table 1: Mean and standard deviation of the F1 score across 6 folds for the single-instance baseline and MIML$_{\text{RoBERTa}}$, with and without hierarchical loss. *random* refers to an ablation where we randomly shuffle the nodes in the hierarchy.

Table 1 shows the micro-averaged F1 scores of the different models discussed in Section 3. We see a significant improvement when using the MIML framework: in cross validation, MIML$_{\text{RoBERTa}}$ has a micro-F1 of 72.10%, an absolute improvement of 1.98% compared to the baseline single-instance model (70.12%). This improvement holds whether we use the hierarchical loss or not – the hierarchical MIML$_{\text{RoBERTa}}$ has a 2.1% improvement over the hierarchical baseline (70.49%).

Method		Tree-F1(%)	
		(Incorrect)	(All)
Baseline	non-hier	51.19	85.44
	hier	51.71	85.78
MIML$_{\text{RoBERTa}}$	non-hier	51.33	86.53
	hier	51.64	86.76

Table 2: Tree-F1 scores for incorrect predictions and all predictions across different models. Incorporating loss$_{\text{aux}}$ improves the Tree-F1 score in both cases.

We also observe small but consistent improvements when training and predicting with hierarchical information in the form of auxiliary classifiers: the baseline model improves by 0.37%, while MIML$_{\text{RoBERTa}}$ improves by 0.49%. Table 2 shows the Tree-F1 scores over the full validation splits, as well as only for the incorrect predictions from the various models. Again, we observe a small improvement due to incorporating the hierarchical information – specifically, the incorrect predictions are now closer to the ground truth labels, as evidenced by the higher Tree-F1 over the mistakes.

To further confirm that these improvements are due to learning hierarchical information rather than any regularization effect from the additional auxiliary loss term, we also run the experiments with the labels on the decision tree randomly shuffled with the same lambda values. We obtain a micro-F1 of 69.54 with $\lambda_{\text{training}} = \lambda_{\text{eval}} = 0.5$, which is significantly lower. Particularly, when $\lambda_{\text{training}} = 0.5$ and $\lambda_{\text{eval}} = 1$, i.e. inference is done using only the auxiliary classifiers, the micro-F1 in cross validation is an extremely low 5.25%, in contrast to a 72.33% when not shuffled.

System		**Test F1**
Single	Singh et al. (2020)	58.436
	Dimov et al. (2020)	59.832
	non-Hier MIML$_{\text{RoBERTa}}$ (Ours)	62.179
	Hier MIML$_{\text{RoBERTa}}$ (Ours)	62.793
Ensemble	Morio et al. (2020)	63.129
	Chernyavskiy et al. (2020)	63.296
	Jurkiewicz et al. (2020)	63.743

Table 3: F1 score on the official test set of various systems on the leaderboard. See https://propaganda.qcri.org/ptc/leaderboard.php

The micro-F1 results on the official test set are in Table 3, where we see that the hierarchical MIML$_{\text{RoBERTa}}$ is the best performing single model. Other than improving accuracy, MIML$_{\text{RoBERTa}}$ also reduces the training and evaluation time, since it predicts multiple spans in a single forward pass through the RoBERTa model. One epoch of MIML$_{\text{RoBERTa}}$ (one evaluation every 25 steps) takes 5.04 minutes compared to 15.50 minutes by the single-instance baseline, a 68% reduction.

6 Conclusion

We propose two simple extensions to a RoBERTa-based model for propaganda technique classification, which lead to notable improvements. Our approach to incorporating hierarchical information about the labels into training could also be useful for other tasks where the annotation procedure involves making a series of decisions about instances. Future work can also explore other methods to incorporate the hierarchical information, e.g., via regularizing the label embeddings.

Limitations

The use of auxiliary classifiers at every node of the decision tree is not feasible when the hierarchical tree is huge, such as the large hierarchical terminologies for medical literature indexing (Gasco et al., 2021).

Besides, in Table 1, even though the integration of the hierarchical information shows a consistent improvement in both the baseline and MIML$_{\text{RoBERTa}}$ models, these improvements are still within one standard deviation of micro-F1.

Lastly, it is worth noting that we do not focus on large language models since our approach is to explore improvements on a published state-of-the-art model. While they might improve accuracy, a careful exploration of those on a new task is beyond the scope.

Ethics Statement

Propaganda detection is a sensitive topic and any practical application of the model needs to be carefully orchestrated. Both false positives and false negatives of the model can have harmful impacts. Moreover, there might be certain biases in the training data, and consequently this leads to systematic issues in the model, such as a higher tendency to mislabel certain kinds of text.

Furthermore, this paper follows the same definition as SemEval 2020 Task 11 (Martino et al., 2020a) whereas there might a broader debate on the definition of propaganda.

Lastly, we also acknowledge the concern that a perfect classifier for propaganda text can be used to train a language model that generates propaganda which in turn evades the classifier's detection.

References

Alberto Barrón-Cedeno, Giovanni Da San Martino, Israa Jaradat, and Preslav Nakov. 2019. Proppy: A system to unmask propaganda in online news. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9847–9848.

Anton Chernyavskiy, Dmitry Ilvovsky, and Preslav Nakov. 2020. Aschern at semeval-2020 task 11: It takes three to tango: Roberta, crf, and transfer learning. *arXiv preprint arXiv:2008.02837*.

Giovanni Da San Martino, Alberto Barrón-Cedeño, and Preslav Nakov. 2019. Findings of the NLP4IF-2019 shared task on fine-grained propaganda detection. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship,* *Disinformation, and Propaganda*, pages 162–170, Hong Kong, China. Association for Computational Linguistics.

Ivica Dimitrovski, Dragi Kocev, Suzana Loskovska, and Saso Dzeroski. 2011. Hierarchical annotation of medical images. *Pattern Recognit.*, 44:2436–2449.

Ilya Dimov, Vladislav Korzun, and Ivan Smurov. 2020. NoPropaganda at SemEval-2020 task 11: A borrowed approach to sequence tagging and text classification. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1488–1494, Barcelona (online). International Committee for Computational Linguistics.

Susan Dumais and Hao Chen. 2000. Hierarchical classification of web content. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, page 256–263, New York, NY, USA. Association for Computing Machinery.

Luis Gasco, Anastasios Nentidis, Anastasia Krithara, Darryl Estrada-Zavala, Renato Toshiyuki Murasaki, Elena Primo-Peña, Cristina Bojo Canales, Georgios Paliouras, Martin Krallinger, et al. 2021. Overview of bioasq 2021-mesinesp track. evaluation of advance hierarchical classification techniques for scientific literature, patents and clinical trials. CEUR Workshop Proceedings.

Monika Glowacki, Vidya Narayanan, Sam Maynard, Gustavo Hirsch, Bence Kollanyi, Lisa-Maria Neudert, Phil Howard, Thomas Lederer, and Vlad Barash. 2018. News and political information consumption in mexico: Mapping the 2018 mexican presidential election on twitter and facebook. *The Computational Propaganda Project*.

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. *arXiv preprint arXiv:2005.07934*.

Aris Kosmopoulos, Ioannis Partalas, Eric Gaussier, Georgios Paliouras, and Ion Androutsopoulos. 2015. Evaluation measures for hierarchical classification: a unified view and novel approaches. *Data Mining and Knowledge Discovery*, 29(3):820–865.

Yuncong Li, Cunxiang Yin, Sheng-hua Zhong, and Xu Pan. 2020. Multi-instance multi-label learning networks for aspect-category sentiment analysis. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3550–3560, Online. Association for Computational Linguistics.

G Martino, Alberto Barrón-Cedeno, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020a. Semeval-2020 task 11: Detection of propaganda techniques in news articles. *arXiv preprint arXiv:2009.02696*.

Giovanni Da San Martino, Stefano Cresci, Alberto Barrón-Cedeño, Seunghak Yu, Roberto Di Pietro, and Preslav Nakov. 2020b. A survey on computational propaganda detection. *arXiv preprint arXiv:2007.08024*.

Gaku Morio, Terufumi Morishita, Hiroaki Ozaki, and Toshinori Miyoshi. 2020. Hitachi at SemEval-2020 task 11: An empirical study of pre-trained transformer family for propaganda detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1739–1748, Barcelona (online). International Committee for Computational Linguistics.

Mayank Raj, Ajay Jaiswal, Rohit R.R, Ankita Gupta, Sudeep Kumar Sahoo, Vertika Srivastava, and Yeon Hyang Kim. 2020. Solomon at SemEval-2020 task 11: Ensemble architecture for fine-tuned propaganda detection in news articles. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1802–1807, Barcelona (online). International Committee for Computational Linguistics.

Carlos N. Silla and Alex A. Freitas. 2011. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1):31–72.

Paramansh Singh, Siraj Sandhu, Subham Kumar, and Ashutosh Modi. 2020. newsSweeper at SemEval-2020 task 11: Context-aware rich feature representations for propaganda classification. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1764–1770, Barcelona (online). International Committee for Computational Linguistics.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 455–465.

Shubo Tian and Jinfeng Zhang. Team fsu2021 at biocreative vii litcovid track: Bert-based models using different strategies for topic annotation of covid-19 literature. In *Proceedings of the seventh BioCreative challenge evaluation workshop*.

Shubo Tian and Jinfeng Zhang. 2021. Multi-label topic classification for covid-19 literature annotation using an ensemble model based on pubmedbert. *bioRxiv*.

Andreas S. Weigend, Erik D. Wiener, and Jan O. Pedersen. 1999. Exploiting hierarchy in text categorization. *Information Retrieval*, 1(3):193–216.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Zhi-Hua Zhou, Min-Ling Zhang, Sheng-Jun Huang, and Yu-Feng Li. 2012. Multi-instance multi-label learning. *Artificial Intelligence*, 176(1):2291–2320.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

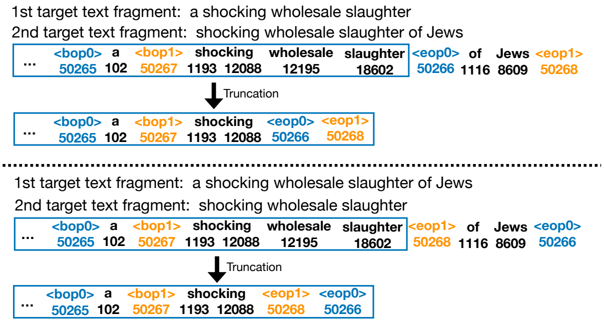
1st target text fragment: a shocking wholesale slaughter
2nd target text fragment: shocking wholesale slaughter of Jews
... <bop0> a <bop1> shocking wholesale slaughter <eop0> of Jews <eop1>
50265 102 50267 1193 12088 12195 18602 50266 1116 8609 50268
Truncation
... <bop0> a <bop1> shocking <eop0> <eop1>
50265 102 50267 1193 12088 50266 50268
1st target text fragment: a shocking wholesale slaughter of Jews
2nd target text fragment: shocking wholesale slaughter
... <bop0> a <bop1> shocking wholesale slaughter <eop1> of Jews <eop0>
50265 102 50267 1193 12088 12195 18602 50268 1116 8609 50266
Truncation
... <bop0> a <bop1> shocking <eop1> <eop0>
50265 102 50267 1193 12088 50268 50266

Figure 3: Illustration of how the truncation respects the nesting structure. The difference in text fragments (above and below the dotted line) is exemplified in different nesting structures before truncation. The difference is also preserved after truncation by the differences in the ordering of <eop0> and <eop1>.

A Techniques

The hierarchical diagram to guide the annotation of propaganda techniques can be found in Figure 4.

B Hyperparameters

We use HuggingFace's library (Wolf et al., 2019) and a single Nvidia RTX A6000 GPU. Both the baseline and the hierarchical baseline methods use a learning rate of $2e - 5$, a dropout of 0 and a batch size of 16, while the MIML$_{\text{RoBERTa}}$ and its hierarchical version use a learning rate of $1e - 5$, a dropout of 0.1 and a batch size of 8. All the models are trained for 20 epochs. More details can be found in Table 4.

C Tree-F1 Metric

Let Y be the ground truth label and Y' be the prediction, and let K be the lowest common ancestor between the two. Then the Tree-F1 is given by:

$$\text{Precision}_{\text{Tree}} = L_K / L_{Y'}$$

$$\text{Recall}_{\text{Tree}} = L_K / L_Y$$

$$\text{Tree-F1} = \frac{2 * \text{Precision}_{\text{Tree}} * \text{Recall}_{\text{Tree}}}{\text{Precision}_{\text{Tree}} + \text{Recall}_{\text{Tree}}}$$

where L_M refers to the number of nodes tracing from the root node down to node M.

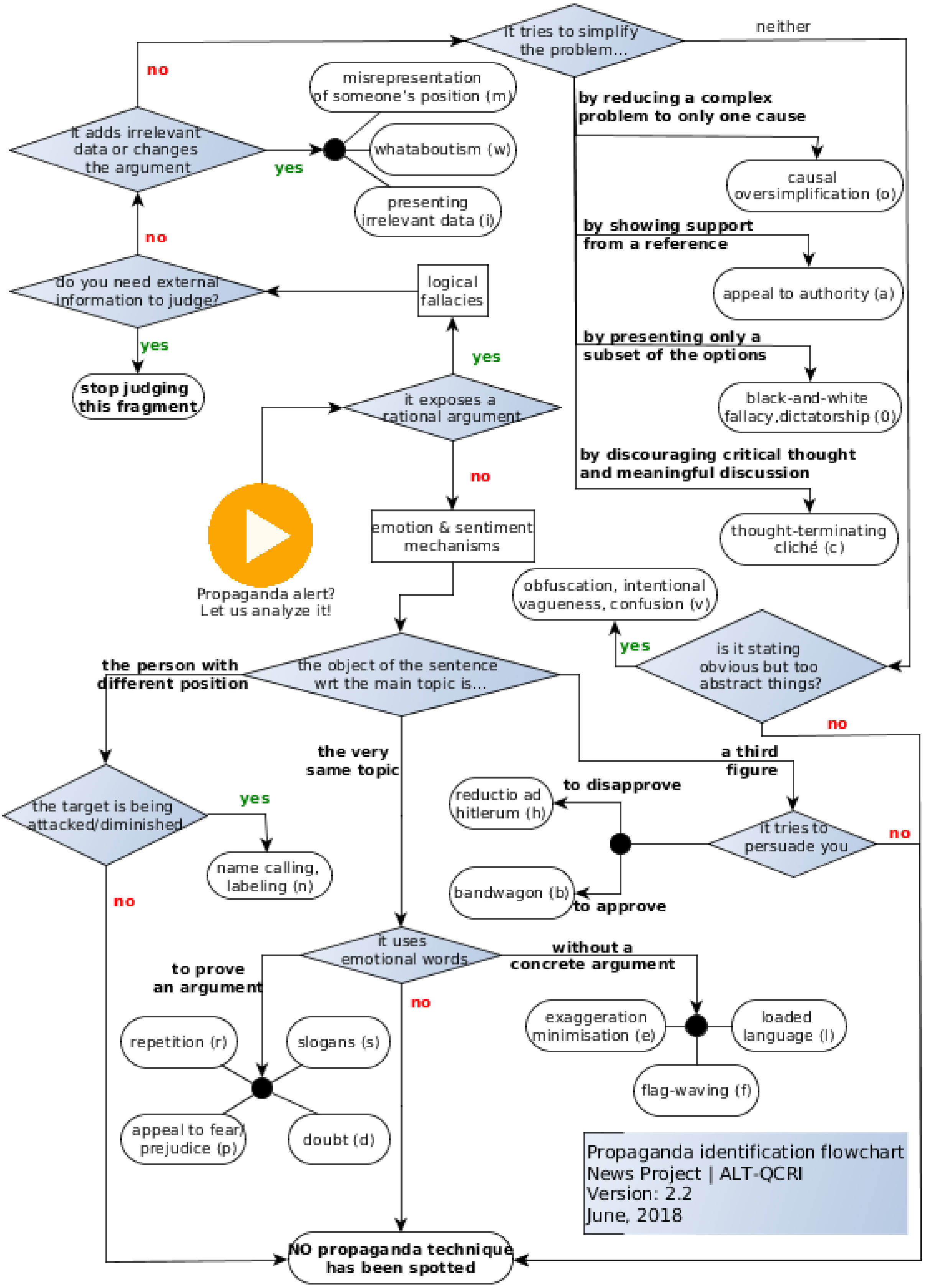
It tries to simplify the problem...
neither
no
misrepresentation of someone's position (m)
It adds irrelevant data or changes the argument
yes
whataboutism (w)
presenting irrelevant data (i)
by reducing a complex problem to only one cause
causal oversimplification (o)
no
by showing support from a reference
appeal to authority (a)
do you need external information to judge?
logical fallacies
yes
by presenting only a subset of the options
yes
stop judging this fragment
it exposes a rational argument
black-and-white fallacy,dictatorship (0)
by discouraging critical thought and meaningful discussion
no
emotion & sentiment mechanisms
thought-terminating cliché (c)
Propaganda alert? Let us analyze it!
obfuscation, intentional vagueness, confusion (v)
yes
is it stating obvious but too abstract things?
the person with different position
the object of the sentence wrt the main topic is...
no
the very same topic
a third figure
yes
to disapprove
reductio ad hitlerum (h)
the target is being attacked/diminished
It tries to persuade you
no
name calling, labeling (n)
no
bandwagon (b)
to approve
it uses emotional words
without a concrete argument
to prove an argument
no
repetition (r)
slogans (s)
exaggeration minimisation (e)
loaded language (l)
flag-waving (f)
appeal to fear/ prejudice (p)
doubt (d)
Propaganda identification flowchart
News Project | ALT-QCRI
Version: 2.2
June, 2018
NO propaganda technique has been spotted

Figure 4: The hierarchical diagram to guide the annotation of propaganda techniques (Martino et al., 2020a).

Hyperparameter	Baseline	Hier Baseline	MIML$_{\text{RoBERTa}}$	Hier MIML$_{\text{RoBERTa}}$
Dropout	0	0	0.1	0.1
Learning Rate	2e-5	2e-5	1e-5	1e-5
Weight decay	0.01	0.01	0.1	0.1
Loss	BCE	BCE	CE	CE
Batch size	16	16	8	8
Context Size	512	512	512	512
Number of epochs	20	20	20	20
Optimizer	AdamW	AdamW	AdamW	AdamW

Table 4: Optimizers and hyperparameters for all different methods.

Technique	**Freq**
Loaded Language	2448
Name calling or labeling	1241
Repetition	766
Exaggeration or minimization	534
Doubt	559
Appeal to fear/prejudice	338
Flag-waving	316
Causal oversimplification	227
Slogans	169
Appeal to authority	158
Black-and-white fallacy, dictatorship	129
Thought-terminating cliché	93
Whataboutism, straw man, red herring	136
Bandwagon, reductio ad hilterum	77

Table 5: Summary of techniques and their frequency in the data. The definitions of the techniques are found in https://propaganda.qcri.org/annotations/definitions.html