UNTLing at SemEval-2020 Task 11: Detection of Propaganda Techniques in News Articles

Maia Petee and **Alexis Palmer**
The University of North Texas
Denton, Texas
maiapetee@my.unt.edu, alexis.palmer@unt.edu

Abstract

Our system, submitted as part of Task 11 of SemEval-2020, explores the ability of semantic-level features to detect and label propagandistic rhetorical techniques in news articles. For the second subtask in this initiative, which seeks to label identified propagandistic fragments with one of fourteen technique labels, our system attained a micro-averaged F1 score of 0.40; in this paper, we take a detailed look at these fourteen labels and the ability of our semantically-focused features to detect each of them. We also propose strategies to fill the gaps left by our system.

1 Introduction

This paper describes our contribution to the shared Propaganda Evaluation (PropEval) task, which is one of this year's SemEval-2020 initiatives. This UNTLing team system, which uses a CRF sequence labeler and a Logistic Regression classifier to address the problems posed by two linked subtasks — respectively the detection of variable-length news article spans that employ propagandistic rhetorical techniques and the 14-way labeling of these identified spans — focuses on attaining blanket performance across the given fourteen-label ontology through the use of semantically-driven features.

The value of a system that can detect and label propaganda when it is used in textual media intended for public use is significant. The amount of text available in our vast global landscape, both content that is created by companies and presented to the public and user-generated content, has grown far beyond the reach of human moderation. The quantity of available information is greatly in excess of human ability to moderate or digest, and this necessarily calls for technology that is able to flag text in which content creators use misleading rhetorical techniques to promote a partisan political or social agenda. Currently, systems like this are rare, and to our knowledge there are none in wide use. While Proppy(Barrón-Cedeno et al., 2019a), the 2019 system that inspired this shared task (see 2 for more information on genesis of this task) has been made available online, its interface is not intuitive to use, and it does not present information in an accessible or informative way. It also does not integrate with the content a user is viewing. Ideally, in the future, propaganda-detection systems and other computational systems like those created in the FEVER[1] shared task will be adapted to be browser-based and interactive; this would give them the ability to combat the frequent and lazy use of propagandistic rhetorical techniques or outright falsehoods in online news and social media. Accessibility to systems that could act not only as fact-checkers but as bulwarks against misleading content in real time would no doubt prove to be a good influence on online content creators; in an ideal scenario, systems like this could influence rhetorical trends in the presentation of online news to skew away from the unnecessarily sensational and toward the factual. In essence, the automated detection of online propaganda could reduce the quantity of that propaganda and promote the creation of a better-informed public.

This system approaches the task of propaganda detection primarily from a semantic perspective. The intuition behind this approach is that semantic information, whether encoded in word embeddings or in sentence-level embeddings, captures a surprising level of linguistic nuance (Chen et al., 2013).

While we worked on both subtasks, we focused primarily on the second subtask (Technique Classification, or TC) and made an official submission of our TC system. Our Subtask 1 system achieved a

[1] http://fever.ai/

Label	Precision	Recall	F1 Score
N	0.992	1.000	0.996
B-P	0.167	0.014	0.025
I-P	0.280	0.014	0.028

Table 1: Results for Subtask 1 (Span Identification).

macro-averaged F1 score of 0.349, but only after the end of the official submission period. Our system for Subtask 2 came in 27th out of 32 teams on the test set (F = 0.391) and 34th out of 42 teams on the development set (F = 0.409). More detailed results are available in Tables 1 and 3.

2 Background

This task came about as a result of years of work on the part of researchers Alberto Barrón-Cedeño from the University of Bologna and Giovanni Da San Martino from the Qatar Computing Research Institute, among others. In 2019, Barrón-Cedeño and Da San Martino developed a coarse-grained machine learning system that they dubbed Proppy (Barrón-Cedeno et al., 2019a); in its initial incarnation, Proppy rated a document's overall propagandistic content using a maximum entropy classifier and returning a probabilistic value along a 0-1 scale. Proppy was published[2] with a user-friendly UI that distills the last 24 hours' news into events, using an event detection and organization system that automatically resolves multiple articles describing the same event into one instance (Barrón-Cedeno et al., 2019b). All the articles identified as describing this event can then be filtered according to their propagandistic content.

A second phase of Proppy's development involved expansion to a multi-granular approach; this allowed the machine learning system to perform propaganda detection on the sentence and even fragment (here, a series of words of undetermined length) level (Da San Martino et al., 2019). This token-level analytical capability enabled the construction of a related tool[3] that visualizes the number of tokens containing propagandistic techniques as compared to the total number of tokens devoted to any particular topic (e.g., gun control). This multi-granular system was the one that was used as a seminal submission. Simple baselines were created by the task organizers and made available to participants to demonstrate the format that scorable output should take.

Task data, in the form of news articles scraped from the web that include gold labels for the beginning and ending character spans of propagandistic fragments (and, in the case of Subtask 2, gold technique labels), was made available to all task participants. Training data comprised 371 articles, almost all containing more than one propagandistic fragment and most containing several, for a total of 5468 propagandistic spans. Development data contained 940 gold-labeled spans in a total of 75 articles, similarly annotated using character offsets from document start. Input was in tab-delimited form; each line of the input represented one propagandistic span, and contained values for article ID, character offset values (in reference to the article start) for both the beginning and the end of the propagandistic span, and a gold technique label that was only used in Subtask 2. The CRF sequence labeler predicted a label — either "B-P", "I-P", or "N" — for each token in the data.

Tasks could be undertaken separately instead of as a pipeline: for instance, despite Subtask 1 not successfully identifying the majority of the tokens labeled as propaganda, those gold-labeled spans could still be used as input for the labeling of Subtask 2. We wrote the output for Subtask 2 to a template file released by the task organizers for the purposes of submission to the task website; the website interface allowed us to view our entire submission history and performance.

3 System Overview

3.1 Subtask 1: Propagandistic Span Identification

For Subtask 1, we use a sequence labeler in the form of a CRF, or conditional random field (Lafferty et al., 2001), algorithm to enable the system to consider contextual factors when identifying spans, which

[2]https://proppy.qcri.org/live
[3]https://www.tanbih.org/propaganda

vary greatly in length from one or two tokens to multi-sentence chunks. As an example, for the following sentence taken from the training data, “Manchin says Democrats acted like babies at the SOTU (video) Personal Liberty Poll Exercise your right to vote” (this sentence is also an example of the slightly “noisy” news article extraction process, given the advertisement in the second half), the sequence labeler would produce the following labels: [“B-P”, “I-P”, “I-P”, “I-P”, “I-P”, “I-P”, “N”, “N”, “N”, “N”, “N”, “N”, “N”, “N”, “N”, “N”]. This classifier labels the first six tokens as propagandistic (with the first indicated as beginning the sequence, and the subsequent five indicated as “inside” a propagandistic sequence) and the remainder as non-. Since no given propagandistic spans cross token boundaries, we use the token as the primary labeling unit; we consider a “B-P” label followed by any number of “I-P” labels as long as no intervening “N” labels were present as a propagandistic fragment.

We use Scikit-Learn[4] for the training and evaluation[5] of our classifiers. In the first phase of development, we use a 70%/30% split of the 371 articles in the training set. In a later phase, we use those 371 articles as training and evaluate the system on the development set of 75 articles. Results for the second configuration are reported in Table 1.

Our system does not deal well with the variability of the target spans’ length and content. The system was unlikely to correctly label a token as “B-P”, especially since “N” was vastly more common as a gold target than were either “B-P” or “I-P,” and thus achieved especially low recall (0.014) compared to its precision (0.167 and 0.280) for each propaganda label “B-P” and “I-P” respectively. This underlabeling of “B-P” tokens consequently led to underlabeling of “I-P” tokens, since without a correctly identified “B-P” token, subsequent “I-P” tokens cannot be identified. As a test to see what role the determination of label boundaries played in the difficulty of this task, we reduced the granularity of the sequence elements to the sentence-fragment level: sequences of several words that were either part of a propagandistic fragment or were not. We broke sentences into “chunks” at both label boundaries and sentence boundaries. We also reduced labels in granularity to “N” and “P,” and used several basic features for this task: the first and last words of each sentence “chunk” and a bag-of-words containing all the words in a fragment, as well as the same information for the two adjacent fragments. We also used vector representations and sentiment scores both for the target chunk and its neighbors as features. This approach achieved phenomenal results: P = 0.94 and R = 0.84 for the fragments deemed to be propaganda, and higher for those non-.

There could be two possible reasons for this: first, that the variable length of spans and asking the system to correctly label an “N” token followed by a “B-P” token or an “I-P” token followed by an “N” token when one of these tokens is a punctuation mark or an innocuous-seeming stopword such as “the” is indeed the most difficult part of a task that necessarily needs to be undertaken at the token level. Another possible reason is that “chunking” the data in this manner helped achieve balance between the very skewed representation of propagandistic fragments and non-.

Final features for Subtask 1 are detailed in Table 2. Several basic features, such as the word itself, its immediate textual context (prevfirstword and nextfirstword), its fine-grained part of speech using the Penn Treebank tagset (Marcus et al., 1993) were included (we retrieved this information as part of spaCy’s NLP pipeline), in addition to the part-of-speech tags of the word’s immediate context. Syntactic dependency information for the target token, a bag-of-words feature consisting of its near context, and the token’s semantic GloVe vectorpennington2014glove was also included. For each token, surrounding tokens (five words on either side, unless token offset from the beginning or end of the article was less than five) were included in two bag-of-words features. The word embeddings of these BoW features were included separately. Finally, we included the word embedding of the target word using a function that passed in each element of the 300-element vector as a separate feature.

3.2 Subtask 2: Technique Classification

Our system for Subtask 2, or Technique Classification, ranked 34th out of 42 competing teams on the development set and 27th out of 32 teams on the test set. Results across all fourteen technique labels are displayed in Table 3.

[4]https://scikit-learn.org/stable/

[5]https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics

Feature	Description
word	Textual representation of token
prevfirstword	Textual representation of previous token
nextfirstword	Textual representation of next token
pos	Fine-grained part-of-speech of token
prevwordpos	Fine-grained part of speech of prevfirstword
nextwordpos	Fine-grained part of speech of nextfirstword
word.dependency	Syntactic dependency of token within sentence
prevword.dependency	Syntactic dependency of prevfirstword
nextword.dependency	Syntactic dependency of nextfirstword
bow_beginning	Bag-of-words preceding token, up to 5 tokens
bow_end	Bag-of-words following token, up to 5 tokens
semantic_left	The vector representation of bow_beginning
semantic_right	The vector representation of bow_end
v1 - v300	300 features, each one element of token's word embedding

Table 2: Features used in final version of Subtask 1

Label	F1-Score (Test)	F1-Score (Development)
Baseline	0.252	0.265
Our System: Weighted Microaverage	**0.391**	**0.409**
Appeal to Authority	0.043	0.083
Appeal to Fear/Prejudice	0.053	0.119
Bandwagon & *Reductio ad hitlerum*	0.000	0.000
Black & White Fallacy	0.000	0.000
Causal Oversimplification	0.029	0.267
Doubt	0.326	0.354
Exaggeration, Minimization	0.118	0.123
Flag-Waving	0.405	0.517
Loaded Language	**0.626***	**0.594***
Name-Calling, Labeling	0.367	0.333
Repetition	0.078	0.118
Slogans	0.176	0.250
Thought-Terminating Cliches	0.000	0.000
Whataboutism, Straw Man, Red Herring	0.000	0.059

Table 3: Results for Subtask 2 (Technique Classification).

This implementation of the Technique Classification task uses a Logistic Regression classifier to label individual propagandistic fragments with one of fourteen technique labels. These technique labels originally formed a taxonomy of 18 labels. Over the course of the task, several labels were combined due to lack of representation in the training data, and other labels, such as Obfuscation, were removed entirely.

The initial baseline implementation of Subtask 2 included only two features: a bag-of-words feature including all the tokens in a given propagandistic fragment, and the word embedding of the fragment. This baseline achieved a micro-averaged F-score of 0.22484. Our final system additionally incorporates named entity (NE) features and features from the VAD Lexicon (see Table 4). The NE features are a list of all NE in the fragment and types of all NE. The VAD Lexicon is a sentiment-analysis dictionary that uses the Valence-Arousal-Dominance model of affect (Warriner et al., 2013) to calculate sentiment of many lexemes in each fragment on several axes rather than a simple positive-negative scale. Sentiment polarity and subjectivity features from spaCy decreased overall performance and were thus not included in the final system.

Many teams in the competition seem to focus closely on maximizing scores for one or two techniques, which, when combined with other systems, could produce a system that excels across all techniques. The goal of our effort is to implement a system that takes into account a variety of sentiment and semantic information: the primary goal is to achieve good results, and this coexists alongside a secondary goal of achieving non-zero results across as many of the propagandistic categories as possible. Largely, we accomplished this: eleven out of fourteen tasks — all except Bandwagon/*Reductio ad Hitlerum* and Thought-Terminating Cliches — achieve some accuracy on the development set (ten on the test set: no Whataboutism/Straw Man/Red Herring fragments are correctly identified for this dataset). It is perhaps worth noting that the system performs the worst on two of the combined labels; we examine these and

Feature	Description
text	Textual representation of all words in fragment
semantic	Propaganda fragment vector
ents	Named entities in fragment
enttypes	Types (PER, LOC, etc.) of named entities in fragment
intensity	# of words found in VAD in each segment divided by fragment length
valence	A list of all valence ratings found in each fragment
arousal	A list of all arousal ratings found in each fragment
dominance	A list of all dominance ratings found in each fragment

Table 4: Features used in final version of Subtask 2

other technique scores more closely in 5.1.

4 Experimental Setup

We processed all articles, as well as the individual fragments in Subtask 2, using spaCy's[6] NLP pipeline. We used this pipeline as a starting point for the extraction of most of the semantic features; the named entity features and all word embeddings are represented in spaCy as token or text span attributes. We represented word embeddings specifically using pretrained GloVe vectors (Chen et al., 2013), (Pennington et al., 2014); since the largest of several English models available via spaCy containa these pretrained embeddings, we use this model to create pipelines via spaCy and extract semantic information. To extract the sentiment features, we created a dictionary and accessed each token's valence, arousal, and dominance values as needed. We also preprocessed our features using a multi-label binarizer in the case of Subtask 1, and a dictionary vectorizer in the case of Subtask 2. Both preprocessors are part of SciKit-Learn [7].

Matplotlib[8] was used to generate confusion matrices.

The task organizers scored the results of individual systems using a micro-averaged F-score.

5 Analysis

In this section we look more closely at the results for both subtasks.

Subtask 1. Here we focus on features that were helpful in enabling the Subtask 1 system to detect and label "B-P" and "I-P" tokens correctly. A feature analysis of the CRF system, using several of Scikit-Learn's evaluation metrics, reveals that the most informative feature is the word embedding of the target token; 10 out of the 30 most informative features are individual token vector elements. Another important set of features concerns the target token's left-hand context: both the bag-of-words before the target token and the vector information of that bag-of-words (bow_beginning and semantic_left) account for 13 out of 30 most informative features. These results are contrasted with their right-hand counterparts (bow_end and semantic_right), which when measured similarly only account for 3 out of 30 informative features.

It seems, then, that the semantic content of tokens that are used in propagandistic rhetoric does differ to some degree from tokens used in non-propagandistic fragments. Additionally, it seems that propaganda is more easily signaled by what comes before it than by what comes after: this could be due to the fact that it is easier for the system to detect a continuation of propagandistic speech (i.e., "I-P" tokens) than it is to detect the critical "B-P" tokens. This is quite probably a result of the vastly skewed distribution of propagandistic tokens to non-; due to this, the system has learned that "B-P" tokens are unlikely to follow "N" tokens in any context, and it learns to favor "N" labels even when semantic information would indicate otherwise.

Subtask 2. The Subtask 2 labeler saw the greatest increase in overall performance when adding two named entity features (ents and enttypes) to two existing semantic features (text and semantic): this caused accuracy on two labels (Flag-Waving and Loaded Language) to shoot from .18 and .24 to .51 and .57,

[6]https://spacy.io/
[7]https://scikit-learn.org
[8]https://matplotlib.org/

	Baseline	Baseline + NE Features	Baseline + VAD Features	Final System
Overall	**0.225**	**0.391**	**0.344**	**0.409**
Appeal to Authority	0.138	0.087	0.000	0.083
Appeal to FearPrejudice	0.071	0.036	0.034	0.119
Bandwagon/Reductio ad Hitlerum	0.000	0.000	0.000	0.000
Black & White Fallacy	0.000	0.000	0.000	0.000
Causal Oversimplification	0.000	0.171	0.077	0.267
Doubt	0.250	0.214	0.295	0.354
Exaggeration/Minimization	0.121	0.027	0.056	0.123
Flag-Waving	0.180	0.506	0.476	0.517
Loaded Language	0.239	0.566	0.539	0.594
Name Calling/Labeling	0.145	0.318	0.286	0.333
Repetition	0.355	0.064	0.118	0.118
Slogans	0.182	0.182	0.255	0.250
Thought-Terminating Clichés	0.000	0.000	0.000	0.000
Whataboutism/Straw Man/Red Herring	0.000	0.056	0.065	0.059

Table 5: Feature ablation (on development set) of Subtask 2 Feature Groups

respectively. The increase in detection of Flag-Waving fragments could be explained by the fact that these fragments, characterized by blind nationalism and almost requiring by definition the inclusion of named entities, do almost always include a reference to the idealized entity ("the UK," "the new Sweden," "Zionism," and numerous uses of "America" and "American"). Conversely, the increase in detection of Loaded Language fragments might be caused by a relative lack of named entities in these fragments that was detected by our classifier. Name Calling/Labeling was another technique whose accuracy was increased by adding named entity features, for perhaps obvious reasons (if a fragment employs the use of pejorative name-calling, it is more likely that the name/label will be a NE).

Three sentiment-related features — valence, arousal, and dominance — boosted the performance of our system on categories related to the cultivation of a particular emotion in the audience, such as Appeal to Fear/Prejudice and Doubt. This makes sense, given that rhetoric designed to instill a particular emotion in others often models that emotion to "set the mood," as in fragments like "warning that an outbreak could occur at any time" and "evolve into something far more dangerous." The sentiments in these fragments have (expert) speculation as supporting evidence, but their vaguely- and anxiously-written premonitions mirror anxiety about worst-case outcomes and do not limit themselves to presenting factual information.

Certain technique labels, such as Appeal to Authority and Repetition, achieved their highest scores on the baseline implementation, highlighting the effectiveness of semantic information for a wide variety of classification tasks. It is also a possibility that selecting features that perform exceedingly well for certain labels, such as Arousal for Loaded Language, has led to the system choosing those labels when another label might be more appropriate.

5.1 Error Analysis

Whatever information was gained by the CRF classifier using our Subtask 1 features, it is still largely insufficient to yield a well-performing system. Our system incorrectly labels "B-P" tokens as "N", and these errors cause a snowball effect that results in a large number of predicted "N" labels for tokens whose true label is "I-P". We discuss a proposal for remedying this pattern in 5.2.

When the Subtask 1 system did predict a "B-P" label, it generally labeled the next 10+ tokens as "I-P." This does not reflect the wide variance in propagandistic span length seen in the gold-labeled data: many consist of one or two tokens (e.g., in the Name-Calling/Labeling category: "futuristic," "despotic leader," etc. and the Loaded Language "very, very").

The Subtask 2 classifier overwhelmingly predicts labels from a small subset (see Figure 1 for details), often incorrectly labeling fragments as Loaded Language and Doubt. When examining the gold-labeled Loaded Language fragments, it becomes clear that this might be due to the wide variety of fragments (both semantically and in terms of length) that are labeled as such. There are many one- and two-word fragments that fit this category, and many others that run for multiple sentences. Additionally, Loaded

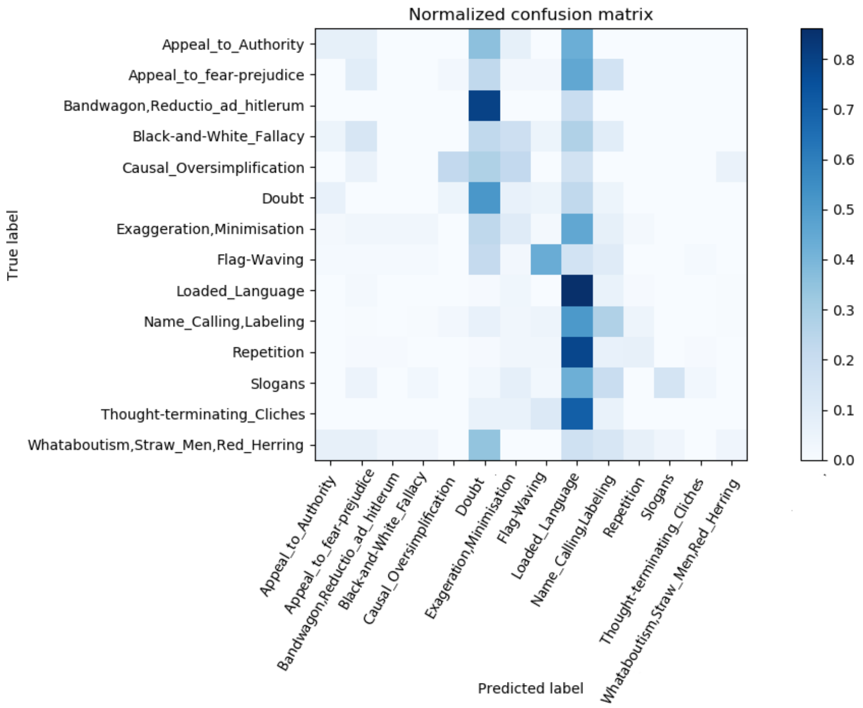

Figure 1: Normalized confusion matrix for Technique Classification

Language seems to be in effect a super-category that prominently overlaps with many other categories, and the classifier might be thwarted by this ambiguity. Many fragments that are labeled with other techniques, such as Exaggeration/Minimization (e.g. "their dream of independence is all but dead"), also use high-affect language that elicits an emotional response in the reader, making it easy to miss pragmatic subtleties and mislabel fragments containing loaded language as belonging to the Loaded Language category.

A similar effect is perhaps in play in the misclassification of many fragments as primarily utilizing Doubt: almost all of the Bandwagon/*Reductio ad Hitlerum* fragments were labeled as Doubt. This could be due to similar sentiment readings across the two labels: both labels are likely to use low-valence (negative) and high-arousal (intense) lexemes. They are also likely to use many named entities of similar categories, given that the goal of both techniques is to cast aspersions on an individual or population.

5.2 Future Iterations

One significant shortcoming of the Subtask 1 implementation is that it did not address the skewed distribution of labels in the training data. At the token level, there is a marked imbalance between the number of propagandistic tokens and the number of non-propagandistic tokens. Roughly 50,000 tokens are labeled as B-P and I-P in the training articles, while almost 12 million tokens are labeled N. Specifically, the ratio of propagandistic tokens to non- is 51186 to 11529733, which means that only .004% of tokens in the training data are propagandistic. The ratio is likely to be similarly skewed in development and testing data. To address this problem in the future, we will downsample the "N" data using an imbalancing package (imbalanced-learn[9]). It is quite possible that downsampling will significantly improve system performance.

Researchers using the gold-labeled data and the official scoring software must also use the official label taxonomy, but it seems as though some of the label combinations only serve to obfuscate critical characteristics of the data: for example, the combination of the previously-separate labels of Bandwagon

[9] https://imbalanced-learn.readthedocs.io/en/stable/

and *Reductio ad Hiterlum*. These two techniques are not similar in presentation; the Reductio ad Hitlerum fragments are semantically distinct from the Bandwagon fragments in that they tend to contain entities known for having been terrorists or terrorized, such as Adolf Hitler, the Jews, the Palestinians, ISIS, Henry VIII, and the Soviet Union. Bandwagon fragments do not share this characteristic. Since the combination of labels seems to have been undertaken for balancing purposes instead of true similarity, our system would in the future separate these combined labels.

Finally, the features our system utilizes are not focused around distinguishing between the pragmatic subtleties inherent to each task. A future round of feature engineering will revolve around finding the differences between frequently misclassified labels and their predicted counterparts, and modeling these to achieve greater coverage among all tasks.

6 Conclusion

Semantic information such as word embeddings, as we have demonstrated, can be used to some small effect to detect and label propagandistic rhetoric in news articles. However, relying upon semantic and sentiment information alone is far from a complete approach to this complex problem. In the future, we will downsample the skewed Subtask 1 data to maximize our classifier's performance. For Subtask 2, we will create a more granular labeling taxonomy and will seek to create semantic- and discourse-level features that distinguish the more specific technique labels (e.g., Thought-Terminating Clichés) from the more general (e.g., Loaded Language). Taking these next steps will ideally help to advance the automatic detection of propaganda in news articles.

References

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

Alberto Barrón-Cedeno, Israa Jaradat, Giovanni Da San Martino, and Preslav Nakov. 2019b. Proppy: Organizing the news based on their propagandistic content. *Information Processing & Management*, 56(5):1849–1864.

Yanqing Chen, Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2013. The expressive power of word embeddings. *arXiv preprint arXiv:1301.3226*.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeno, 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 5640–5650.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. 2013. Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavior research methods*, 45(4):1191–1207.