Duluth at SemEval-2019 Task 4: The Pioquinto Manterola Hyperpartisan News Detector

Saptarshi Sengupta and **Ted Pedersen**
Department of Computer Science
University of Minnesota
Duluth, MN 55812, USA
{sengu059,tpederse}@d.umn.edu

Abstract

This paper describes the Pioquinto Manterola Hyperpartisan News Detector, which participated in SemEval–2019 Task 4. Hyperpartisan news is highly polarized and takes a very biased or one–sided view of a particular story. We developed two variants of our system, the more successful was a Logistic Regression classifier based on unigram features. This was our official entry in the task, and it placed 23rd of 42 participating teams. Our second variant was a Convolutional Neural Network that did not perform as well.

1 Introduction

Social media has become a vital source of news for many people. It makes it possible to share useful information widely and in a timely fashion, and yet can also be misused to spread biased, misleading, or dangerous content.

Hyperpartisan news is a particular worry in that it is premised on absolute allegiance to one particular point of view, and seeks to reinforce potentially misinformed opinions held by its readers. This has led to very real consequences in this world. A tragic example can be found in Myanmar, where Buddhist ultranationalists relied on social media to spread hyperpartisan and fake news in order to promote hatred and violence against different Muslim communities (Fink, 2018).

While related, Hyperpartisan news is not the same as fake news. The former shows a high degree of bias, whereas the latter is more so an outright fabrication. However, the techniques applied to detecting both are similar. For example, Pérez-Rosas et al. (2018) detected fake news by training Support Vector Machines using ngrams, punctuation, and measures of readability. (Tacchini et al., 2017) used *likes* of articles as features for building a Logistic Regression classifier for fake news detection. Potthast et al. (2018) identified hyperpartisan news through the use of style and readability features, and also employed a technique known as *unmasking* (Koppel et al., 2007) to distinguish between hyperpartisan and mainstream news.

2 Task Description

SemEval–2019 Task 4 (Kiesel et al., 2019) challenged participants to detect whether an article is hyperpartisan (H) or mainsteam (M). As such it represents a binary classification task. The task organizers provided training data, and so we elected to take a supervised learning approach.

There were two datasets provided by the organizers (Kiesel et al., 2018). The *by-article* data is a smaller corpus of 645 news articles that have been manually assigned to H (238 articles) or M (407 articles). There was also the much larger *by-publisher* data set with 750,000 articles where classifications were made based on the source of an article. Making classifications in this way is possible since certain publishers are known to be providers of hyperpartisan content. For our experiments we elected to use the by-article data, but plan to investigate the potential of the by-publisher data in future work.

3 Methodology

We created two systems for the task.[1] The first was a Logistic Regression (LR) classifier trained on unigram features, and the second a Convolutional Neural Network (CNN) with word embeddings created from the training data.

During the development phase of our systems we carried out 10-fold cross validation on the *by-article* training data in order to tune both our LR and CNN systems.

[1] https://github.com/saptarshi059/SemEval2k19-Task4-UMD

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 949–953
Minneapolis, Minnesota, USA, June 6–7, 2019. ©2019 Association for Computational Linguistics

3.1 Logistic Regression

Logistic Regression (LR) is a widely used method for supervised learning. Each feature is assigned a positive or negative weight which indicates the contribution of that feature to the overall classification of the system. We carried out our experiments using *scikit-learn* (Pedregosa et al., 2011), a Machine Learning toolkit for Python.

Our first step was to preprocess the text. This consisted of converting all text to lowercase, and removing stopwords and non-alphanumeric characters.

Next, a word by article matrix was generated for the training data. For our purposes words are defined as space separated strings. Any word that occurred less than 12 times in the training data was removed and not considered a feature. We arrived at this cutoff via our cross validation experiments, where this value led to the most accurate results (although other nearby values were nearly as accurate).

Our LR model was trained using the default settings for scikit-learn. We relied on the default *liblinear* algorithm (Fan et al., 2008) to optimize the loss function, since it is known to be effective with smaller amounts of training. data[2]

3.2 Convolutional Neural Network

Our initial focus was on our LR approach. However, the task allowed for two entries per team, and so we decided to include a CNN given its history of success in text classification tasks (e.g., (Liu and Wu, 2018)). We used *keras* (Chollet et al., 2015), a Python toolkit for Deep Learning that provides a wrapper around TensorFlow.

Our CNN approach was also based on unigrams, although each unigram was represented by an embedding created from the training data. We started with an existing CNN for text classification[3] and made a few adjustments to some of the hyperparameters. The maximum input vector size was set to 10,000, our embeddings were of length 100, and we trained our model for 100 epochs. We used *Adam* to optimize the loss function and a *GlobalMaxPooling1D* layer to reduce the size of the input feature vectors.

We did not experiment with these hyperparameters extensively, but instead relied on what we found to be fairly common settings and defaults provided by *keras*.

Model	Accuracy	P	R	F1
LR	0.70	0.74	0.63	0.68
CNN	0.58	0.87	0.18	0.30

Table 1: Final Evaluation Results.

4 Experimental Results

The formal task evaluation was carried on virtual systems provided by the organizers using the TIRA system (Potthast et al., 2019). We trained both our LR and CNN on the entire by-article training corpus and saved the resulting models to disk (so they could be ported over to the evaluation system).

We decided to use LR and CNN as our two entries to the task, since during our development phase they had very similar results on 10-fold cross validation : LR accuracy was 0.77 ± 0.06 while CNN was at 0.75 ± 0.05.

However, on the official evaluation run (using a held out set of test data the systems had never seen), the CNN performed poorly and only attained accuracy of 0.58. LR on the other hand reached accuracy of 0.70 and so was selected by the organizers as our official entry to the task. Other evaluation metrics including Precision (P), Recall (R), and F1 are shown in Table 1.

The confusion matrix for our LR system is shown in Table 2 and for the CNN system in Table 3. In these matrices the distribution of correct or gold standard answers are shown in the columns (with sums 314) and the system predictions are shown across in the rows. While the evaluation phase test data is balanced between the classes H and M, the by-article training data was not (238 H versus 407 M).

Table 2 shows that LR predicted a somewhat more balanced distribution of classes (266 H vs. 362 M), which is reflected in the relatively similar Precision and Recall scores found in Table 1. However, Table 3 shows that the CNN produced a much more skewed result (67 H vs. 561 M) which led to very high Precision for the CNN (0.87) while the Recall was extremely low (0.18).

We hypothesize that the difference between the distribution of classes in the training versus evaluation data at least partially explains this result. Given more examples of mainstream news (M),

[2] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[3] https://realpython.com/python-keras-text-classification/

	H	M	
H	197	69	266
M	117	245	362
	314	314	628

Table 2: LR Confusion Matrix.

	H	M	
H	58	9	67
M	256	305	561
	314	314	628

Table 3: CNN Confusion Matrix.

both models learned this class more thoroughly and so tended to classify articles into this category.

The LR model appears to be more robust in that it performed at approximately the same level of accuracy both during development phase cross validation and the final evaluation round (despite the difference in the distribution of classes).

The CNN on the other hand appears to have been very negatively affected by the shift in the distribution of classes from training to evaluation data, and performed significantly worse on the evaluation data as compared with cross validation on the training set. We are uncertain as to the causes of the CNN result. It is important to note that the by-article data is relatively small and that this may put the CNN at a disadvantage. We also noticed that the accuracy of the CNN on the training data was 1.00 and much lower on the evaluation data, which is a common sign of overfitting.

5 Feature Analysis

An appealing quality of Logistic Regression is that it is somewhat transparent and allows us to see which features are contributing more to classification decisions. Table 4 shows the top 30 features for LR based on the weights learned from the training data. Positive weights are associated with the hyperpartisan (H) class, and negative weights indicate the mainstream (M) class. We've put the words with negative weights in upper case to improve readability, however remember in our data that all text was lower cased.

While the highly weighted individual features are of interest, it is important to remember that Logistic Regression performs classification based on the combined weight of all the features present in an article. As a result a single highly weighted feature for one class may be overridden by the presence of multiple lesser weighted values for the other class.

Hyperpartisan		Mainstream	
sponsored	.603	**DONALD**	-.611
women	.489	ISIS	-.610
americans	.473	TOOK	-.500
change	.471	SATURDAY	-.417
proud	.463	TWITTER	-.414
hillary	.459	THINK	-.393
arpaio	.440	**WATTERS**	-.392
racist	.436	WORLD	-.389
someone	.433	CLAIMS	-.372
outrage	.423	RUN	-.353
mexican	.373	WEDNESDAY	-.351
threat	.371	ASKED	-.348
political	.370	FREEDOM	-.343
democracy	.369	BORDER	-.334
planned	.366	VIDEO	-.333
supremacist	.364	CONVENTION	-.332
clintons	.337	STATES	-.326
department	.335	ELECT	-.321
use	.329	DEBATE	-.321
desperate	.329	PAST	-.320
originally	.329	**SESSIONS**	-.316
killer	.323	MORNING	-.316
certainly	.322	SAID	-.313
conservative	.320	COUNTY	-.311
father	.313	FOX	-.310
fine	.302	ADVERTISEMENT	-.310
hitler	.302	CONTINUE	-.308
wants	.302	UNITED	-.303
maria	.301	BUSINESS	-.303
make	.299	PRISON	-.301

Table 4: Top 30 LR features : positive weights associated with H, negative with M.

The data for this task consists of articles from 2016 – 2018, starting around the time of the 2016 US presidential election, where Donald Trump defeated Hillary Clinton after a bitterly contested campaign.

In general the top features contain many terms associated with elections or political figures. We note a few more person names among the top 30 features for the H class (5) versus the M class (3). These features are in bold face in Table 4. It is significant to note that one of the person names that appears as a hyperpartisan feature is Hitler, suggesting that he may have been used as a basis for comparison in such articles. The name Arpaio refers to a controversial sheriff in Arizona who ran for re-election in 2016 (and was defeated). Maria is Hurricane Maria, which devastated Puerto Rico in September 2017. The recovery from this natural disaster became a political issue and so its use as a feature in hyperpartisan news seems likely.

The mainstream features (in upper case) include Donald and Twitter. Candidate (and now President) Trump is well known as an enthusiastic Twitter user, so these features would certainly occur in mainstream news coverage. Jesse Watters is a Fox News reporter who hosts a person on the street style interview program which drew some news coverage. Jeff Sessions was an early supporter of Donald Trump and became Attorney General after the election and so was often in the news.

6 Error Analysis

We divided the by-article training data into a set of 585 training examples and 60 test instances (30 from each class). We used this data to train and evaluate our LR classifier. We categorized our results as True Positive (H classified as H), True Negative (M classified as M), False Positive (M classified as H), and False Negative (H classified as M). Below we discuss an article from each category, where each is identified via (by-article id number, word count).

True Positive (1, 259): This article takes a mocking and sarcastic tone regarding President Trump's campaign promises to fix infrastructure. It points out that Hurricane Maria (H feature) did extensive damage but that Trump was indifferent because Puerto Rico did not vote for him. This is an obvious example of hyperpartisan news.

True Negative (14, 225): Ivana Trump, Donald's ex-wife, talks about his punctuality in his personal and professional life. The article is very matter of fact and simply describes her observations without embellishment or bias, and is pretty clearly mainstream.

False Positive (4, 929): This is a very long article that was classified as H despite not having any obvious signs of bias. Rather it compares the unsettled state of America now with the very turbulent year of 1968. However, the article uses many rare and emotional words such as *nihilism*, *malady*, and *hysteria* which may have caused it to be classified as hyperpartisan.

False Negative (2, 189): This is a highly opinionated response to Joyce Newman's (Democrat) stance on gun control. It is a very emotional piece, however, it also provides facts and figures to bolster the position of the author. We believe it is the latter which caused the LR to (incorrectly) classify it as mainstream.

We also noticed that the 30 H articles in our test data had on average much larger word counts (1,178.9) versus the 30 M articles (503.4). (Potthast et al., 2018) used average paragraph length as a feature when detecting H news, and this seems like it would have been a useful feature in this task as well.

7 Future Work

There are numerous possible directions for future work. We are interested in exploring the use of the much larger by-publisher training data. This could be of particular assistance in improving the results from CNNs. We also plan to revisit our preprocessing steps and perform named entity recognition since proper nouns represent important information for this problem.

We would also like to explore variations in our feature sets for LR. In our current experiments we do not have any requirement that a feature occur in a certain minimum number of articles (in addition to occurring at least 12 times). As a result we noticed several features that occurred many times in just a few articles were strongly weighted and yet would be unlikely to generalize well. We would also like to explore the use of *TF-IDF* in place of simple frequency counts for feature selection.

Finally, our error analysis suggested that hyperpartisan news tends to use emotional language as well as unusual or rare words. Given this we are interested in the possibilities offered by sentiment analysis, as well as the inclusion of structural and style features.

8 Namesake

Pioquinto Manterola is a fictional journalist created by Paco Ignacio Taibo II. He is a central character in *The Shadow of a Shadow* (Ignacio Taibo II, 1991) and *Returning as Shadows* (Ignacio Taibo II, 2003). These novels are set in Mexico City, the first in 1922 and the second in 1941-1942. In both stories Manterola is teamed with a poet to investigate mysterious circumstances that lead to uncovering even more complex and sinister wrongdoing. As such he seemed an appropriate namesake for our team in this task.

References

François Chollet et al. 2015. Keras. https://keras.io.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Christina Fink. 2018. Dangerous speech, anti-muslim violence, and facebook in myanmar. *Journal of International Affairs*, 71(1.5):43–52.

Paco Ignacio Taibo II. 1991. *The Shadow of the Shadow*. Viking Books, New York City.

Paco Ignacio Taibo II. 2003. *Returning as Shadows*. Thomas Dunne Books, New York City.

Johannes Kiesel, Maria Mestre, Rishabh Shukla, Emmanuel Vincent, Payam Adineh, David Corney, Benno Stein, and Martin Potthast. 2019. SemEval-2019 Task 4: Hyperpartisan News Detection. In *Proceedings of The 13th International Workshop on Semantic Evaluation (SemEval 2019)*. Association for Computational Linguistics.

Johannes Kiesel, Maria Mestre, Rishabh Shukla, Emmanuel Vincent, David Corney, Payam Adineh, Benno Stein, and Martin Potthast. 2018. Data for PAN at SemEval 2019 Task 4: Hyperpartisan News Detection.

Moshe Koppel, Jonathan Schler, and Elisheva Bonchek Dokow. 2007. Measuring differentiability: Unmasking pseudonymous authors. *Journal of Machine Learning Research*, 8:1261–1276.

Yang Liu and Yi-fang Brook Wu. 2018. Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 354–361.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2018. Automatic detection of fake news. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3391–3401. Association for Computational Linguistics.

Martin Potthast, Tim Gollub, Matti Wiegmann, and Benno Stein. 2019. TIRA Integrated Research Architecture. In Nicola Ferro and Carol Peters, editors, *Information Retrieval Evaluation in a Changing World - Lessons Learned from 20 Years of CLEF*. Springer.

Martin Potthast, Johannes Kiesel, Kevin Reinartz, Janek Bevendorff, and Benno Stein. 2018. A Stylometric Inquiry into Hyperpartisan and Fake News. In *56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, pages 231–240. Association for Computational Linguistics.

Eugenio Tacchini, Gabriele Ballarin, Marco L. Della Vedova, Stefano Moret, and Luca de Alfaro. 2017. Some like it hoax: Automated fake news detection in social networks. *CoRR*, abs/1704.07506.