UBC-NLP at SemEval-2019 Task 4: Hyperpartisan News Detection With Attention-Based Bi-LSTMs

Chiyu Zhang **Arun Rajendran** **Muhammad Abdul-Mageed**
Natural Language Processing Lab
The University of British Columbia
chiyu94@alumni.ubc.ca, arun95@math.ubc.ca, muhammad.mageeed@ubc.ca

Abstract

We present our deep learning models submitted to the SemEval-2019 Task 4 competition focused at Hyperpartisan News Detection. We acquire best results with a Bi-LSTM network equipped with a self-attention mechanism. Among 33 participating teams, our submitted system ranks top 7 (65.3% accuracy) on the *labels-by-publisher* sub-task and top 24 out of 44 teams (68.3% accuracy) on the *labels-by-article* sub-task (65.3% accuracy). We also report a model that scores higher than the 8th ranking system (78.5% accuracy) on the *labels-by-article* sub-task.

1 Introduction

Spread of *fake news* (e.g., Allcott and Gentzkow (2017); Horne and Adali (2017)) (or 'low-quality' information (Qiu et al., 2017), among other terms) can have destructive economic impacts (Sandoval, 2008), result in dangerous real world consequences (Akpan, 2016), or possibly undermine the very democratic bases of modern societies (Qiu et al., 2017; Allcott and Gentzkow, 2017). Several approaches have been employed for detecting fake stories online, including detecting the sources that are highly polarized (or *hyperpartisan*)] (Potthast et al., 2017). Detecting whether a source is extremely biased for or against a given party can be an effective step toward identifying fake news.

Most research on news orientation prediction employed machine learning methods based on feature engineering. For example, Pla and Hurtado (2014) use features such as text n-grams, part-of-speech tags, hashtags, etc. with an SVM classifier to tackle political tendency identification in twitter. Potthast et al. (2017) investigate the writing style of hyperpartisan and mainstream news using a random forest classifier (Koppel et al., 2007). Further, Preoţiuc-Pietro et al. (2017) use a linear regression algorithm to categorize Twitter users into a fine-grained political group. The authors were able to show a relationship between language use and political orientation.

Nevertheless, previous works have not considered the utility of deep learning methods for hyperpartisanship detection. Our goal is to bridge this gap by investigating the extent to which deep learning can fare on the task. More precisely, we employ several neural network architectures for hyperpartisans news detection, including long short-term memory networks (LSTM), convolutional neural networks (CNN), bi-directional long short term memory networks (Bi-LSTM), convolutional LSTM (CLSTM), recurrent convolutional neural network (RCNN), and attention-based LSTMs and Bi-LSTMs.

We make the following contributions: (1) we investigate the utility of several deep learning models for classifying hyperpartisan news, (2) we test model performance under a range of training set conditions to identify the impact of training data size on the task, and (3) we probe our models with an attention mechanism coupled with a simple visualization method to discover meaningful contributions of various lexical features to the learning task. The rest of the paper is organized as follows: data are described in Section 2, Section 3 describes our methods, followed by experiments in Section 4. Next, we explain the results in detail and our submission to SemEval-2019 Task4 in Section 4. We present attention-based visualizations in Section 5, and conclude in Section 6.

2 Data

Hyperpartisan news detection is the SemEval-2019 task 4 (Kiesel et al., 2019). The task is set up as binary classification where data released by organizers are labeled with the tagset

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 1072–1077
Minneapolis, Minnesota, USA, June 6–7, 2019.

	Labels-by-Publisher				Labels-by-Article		
	Train	**Dev**	**Test**	**Total**	**Train**	**Test**	**Total**
Hyperpartisan	383,151	66,849	50,000	500,000	214	24	238
Non- Hyperpartisan	416,849	33,151	50,000	500,000	366	41	407
Total	800,000	100,000	100,000	1,000,000	580	65	645

Table 1: Distribution of labels over our data splits.

{*hyperpartisan, not-hyperpartisan*}. The dataset has two parts, pertaining how labeling is performed. For **Part 1: labels-by-publisher**, labels are propagated from the publisher level to the article level. Part 1 was released by organizers twice. First 1M articles (less clean) were released, but then 750K (cleaner, de-duplicated) articles were released. We use all the 750K articles but we also add 250K from the first release, ensuring there are no duplicates in the articles and we also perform some cleaning of these additional 250K articles (e.g., removing error symbols). We ensure we have the balanced classes {hyperpartisan, not-hyperpartisan}, with 500K articles per class. For experiments, we split Part 1 into 80% train, 10% development (dev), and 10% test.

The labeling method for Part 1 assumes all articles by the same publisher will reflect the publisher's same polarized category. This assumption is not always applicable, since some articles may not be opinion-based. For this reason, organizers also released another dataset, **Part 2: labels-by-article**, where each individual article is assigned a label by a human. Part 2 is smaller, with only 645 articles (238 hyperpartisan and 407 non-hyperpartisan). Since Part 2 is smaller, we split it into 90% train and 10% test. Since we do not have a dev set for Part 2, we perform all our Hyperparameter tuning on the Part 1 dev set exclusively. Table 1 shows the statistics of our data.

3 Methods

3.1 Pre-processing

We lowercase all the 1M articles, tokenize them into word sequences, and remove stop words using *NLTK* [1]. For determining parameters like maximum sequence length and vocabulary size, we analyze the 1M articles, and find the number of total tokens to be 313,257,392 and the average length of an article to be 392 tokens (with a standard deviation of 436 tokens), and the number of types (i.e., unique tokens) to be 773,543. We thus set the maximal length of sequence in our models to be 392, and choose an arbitrary (yet reasonable) vocabulary size of 40,000 words.

[1] https://www.nltk.org/

3.2 Architectures

Deep learning has boosted performance on several NLP tasks. For this work, we experiment with a number of methods that have successfully been applied to text classification. Primarily, we employ a range of variations and combinations of recurrent neural networks (RNN) and convolutional neural networks (CNN). RNNs are good summarizers of sequential information such as language, yet suffer from gradient issues when sequences are very long. Long-Short Term Memory networks (LSTM) (Hochreiter and Schmidhuber, 1997) have been proposed to solve this issue, and so we employ them. Bidirectional LSTM (Bi-LSTM) where information is summarized from both left to right and vice versa and combined to form a single representation has also worked well on many tasks such as named entity recognition (Limsopatham and Collier, 2016), but also text classification (Abdul-Mageed and Ungar, 2017; Elaraby and Abdul-Mageed, 2018). As such, we also investigate Bi-LSTMs on the task. Attention mechanism has also been proposed to improve machine translation (Bahdanau et al., 2014), but was also applied successfully to various other tasks such as speech recognition, image captioning generation, and text classification (Xu et al., 2015; Chorowski et al., 2015; Baziotis et al., 2018; Rajendran et al., 2019). We employ a simple attention mechanism (Zhou et al., 2016b) to the output vector of the (Bi-)LSTM layer. Although CNNs have initially been proposed for image tasks, they have also been shown to work well for texts (e.g., (Kim, 2014)) and so we employ a CNN. In addition, neural network architectures that combine different neural

network architectures have shown their advantage in text classification (e.g., sentiment analysis). For example, improvements on text classification accuracy were observed applying a model built on a combination of Bi-LSTM and two-dimensional CNN (2DCNN) compared to separate RNN and CNN models (Zhou et al., 2016a). Moreover, a combination of CNN and LSTM (CLSTM) outperform both CNN and LSTM on sentiment classification and question classification tasks (Zhou et al., 2015). The experiments of Lai et al. (2015) demonstrate that recurrent convolutional neural networks (RCNNs) outperforms CNN and RNN on text classification. For these reasons, we also experiment with RCNN and CLSM.

3.3 Hyper-Parameter Optimization

For all our models, we use the top 40K words from Part 1 training set (*labels-by-publisher*) as our vocabulary. We initialize the embedding layers with Google News Word2Vec model. [2] For all networks, we use a single hidden layer. We use dropout (Srivastava et al., 2014) for regularization.

Models	Hidden No.	Drop out	Kernel size	Kernel No
LSTM	300	0.1	N/A	N/A
Bi-LSTM	200	0.0	N/A	N/A
LSTM+Attn	500	0.0	N/A	N/A
Bi-LSTM+Attn	500	0.0	N/A	N/A
CNN	N/A	0.1	[4,5,6]	200
RCNN	200	0.3	N/A	N/A
CLSTM	200	0.3	[2,3,4]	70

Table 2: Our best Hyper-parameters.

For the best Hyper-parameters for each network, we use the Part 1 dev set to identify the number of units (between 100 and 600) in each network's hidden layer and the dropout rate (choosing values between 0 and 1, with 0.1 increments). For the CNNs (and their variations), we use 3 kernels with different sizes (with groups like 2,3,4) and identify the best number of kernel filters (between 30 to 300). All Hyper-parameters are identified using the Part 1 dev set. Table 2 presents the detailed optimal Hyper-parameters for all our models. [3]

[2] https://github.com/mmihaltz/word2vec-GoogleNews-vectors

[3] For all our networks, we identify our best learning rate as 0.001. For this reason, we do not provide learning rate in Table 2.

4 Experiments & Results

We run two main sets of experiments, which we will refer to as EXP-A and EXP-B. For EXP-A, we train on the labels-by-publisher (Part 1) train set, tune on dev, and test on test. All related results are reported in Table 3. As Table 3 shows, our best macro F_1 as well as accuracy is acquired with Bi-LSTM with attention (Bi-LSTM+ATTN). For EXP-B, we use Part 1 and Part 2 datasets in tandem, where we train on each train set independently and (1) test on its test data, but also (2) test on the other set's test data. We also (3) fine-tune the models pre-trained on the bigger dataset (Part 1) on the smaller dataset (Part 2), to test the transferrability of knowledge from these bigger models. Related results (only in accuracy, for space) are in Table 4. Again, the best accuracy is obtained with Bi-LSTM with attention.

SemEval-2019 Task 4 Submissions: We submitted our Bi-LSMT+Attention model from EXP A to the *labels-by-publisher leaderboard* in TIRA (Potthast et al., 2019), and it ranked top 7 out of the 33 teams, scoring at *accuracy=0.6525* on the competition test set. [4] From EXP-B, we submitted our model based on Bi-LSMT+Attention that was trained on Part 2 train exclusively dataset (by-ATC in Table 4) to the *labels-by-article leaderboard*. It ranked top 24th out of 44 teams (*accuracy=0.6831*). Post-competition, we submitted our EXP-B model that is pre-trained on the by-publisher data and fine-tuned on the by-article data (by-PSH+by-ATC in Table 4) to the *labels-by-article leaderboard*. It ranked top 8th, with *78.50% accuracy*. This might be due to the ability of this specific model to transfer knowledge from the big (*by-publisher*) training set to the smaller (*by-article*) data (i.e., better generalization).

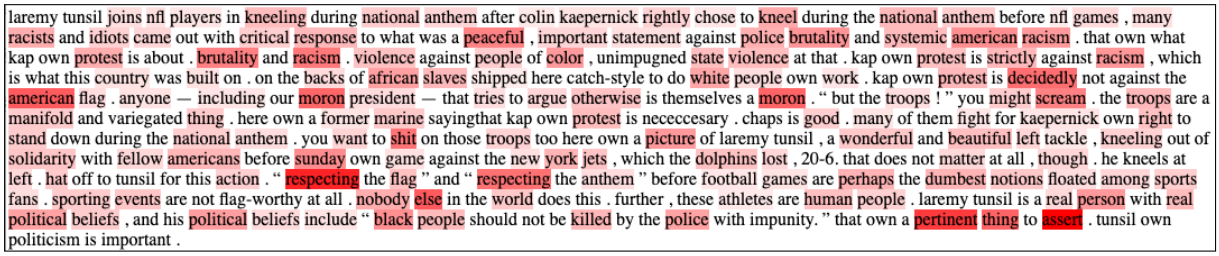
5 Attention Visualization

For better interpretation, we present a visualization of words of our best model from EXP-B (by-PSH+by-ATC in Table 4) attends to across the two classes, as shown in Figure 1. The color intensity in the Figure corresponds to the weight given to each word by the self-attention mechanism and signifies the importance of the word for final prediction. As shown in Figure 1 (a), some heavily polarized terms such as 'moron', 'racism', 'shit',

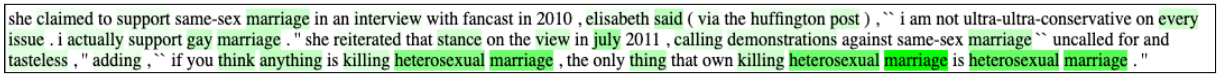
[4] The competition test set is different from our own test set, which we created by splitting the data we received.

Models	Test Accuracy	Precision		Recall		F_1	
		Hyper	Non-hyper	Hyper	Non -hyper	Hyper	Non-hyper
LSTM	0.9174	0.8927	0.9422	**0.9392**	0.8977	0.9154	0.9203
CNN	0.9147	0.9179	0.9115	0.9121	0.9173	0.9150	0.9114
Bi-LSTM	0.9196	0.9097	0.9295	0.9281	0.9114	0.9188	0.9203
LSTM+ATTN	0.9071	0.8755	0.9388	0.9347	0.8829	0.9041	0.9100
Bi-LSTM+ATTN	**0.9368**	**0.9493**	0.9262	0.9347	**0.9480**	**0.9376**	**0.9360**
CLSTM	0.8977	0.9181	0.8773	0.9147	0.8821	0.8956	0.8998
RCNN	0.9161	0.9380	0.8946	0.8972	0.9364	0.9171	0.9150
Random Forest	0.7723	0.5312	**0.9456**	0.8824	0.7333	0.6628	0.8260

Table 3: Performance of Predicting Hyperpartisan News (EXP-A).



(a) Hyperpartisan.



(b) Non-hyperpartisan.

Figure 1: Attention heat-map for article examples.

	Test on	Train on		
		by-PSH	by-ATC	by-PSH +by-ATC
LSTM	**by-PSH**	0.9174	0.5331	0.8369
	by-ATC	0.5917	0.7833	0.7667
BiLSTM	**by-PSH**	0.9196	0.5562	0.8089
	by-ATC	0.5783	0.6540	0.7833
LSTM+A	**by-PSH**	0.9071	0.7397	0.8509
	by-ATC	0.5783	0.8166	0.7833
BiLSTM+A	**by-PSH**	**0.9368**	0.5412	0.7908
	by-ATC	0.5504	**0.8615**	**0.8153**

Table 4: Results with Part 1 and Part 2 datasets (EXP-B). Last column “by-PSH +by-ATC” is the setting of our models pre-trained on Part 1 and fine-tuned on Part 2. +A= added attention.

‘scream’, and ‘assert’ are associated with the hyperpartisan class. It is clear from the content of the article from which the example is drawn that it is a highly opinionated article. In Figure 1 (b), items such as ‘heterosexual marriage’, ‘gay’, ‘July’, and ‘said’ carry more weight than other items. These items are not as much opinionated as those in 1 (a), and some of them (e.g., ‘July’ and ‘said’) are more of factual and reporting devices than mere carriers of ad hominem attacks. These features show that some of the model attentions are meaningful.

6 Conclusion

In this paper, we described our system of hyperpartisan news detection to the 4th SemEval-2019 shared task. Our best models are based on a Bi-LSTM with self-attention. To understand our models, we also visualize their attention weights and find meaningful patterns therein.

7 Acknowledgement

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), the Social Sciences Research Council of Canada (SSHRC), WestGrid (www.westgrid.ca), and Compute Canada (www.computecanada.ca).

References

Muhammad Abdul-Mageed and Lyle Ungar. 2017. Emonet: Fine-grained emotion detection with gated recurrent neural networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 718–728.

Nsikan Akpan. 2016. The very real consequences of fake news stories and why our brain cant ignore them. *PBS News Hour*.

Hunt Allcott and Matthew Gentzkow. 2017. Social media and fake news in the 2016 election. Technical report, National Bureau of Economic Research.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Christos Baziotis, Nikos Athanasiou, Alexandra Chronopoulou, Athanasia Kolovou, Georgios Paraskevopoulos, Nikolaos Ellinas, Shrikanth Narayanan, and Alexandros Potamianos. 2018. Ntua-slp at semeval-2018 task 1: Predicting affective content in tweets with deep attentive rnns and transfer learning. *arXiv preprint arXiv:1804.06658*.

Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-based models for speech recognition. In *Advances in neural information processing systems*, pages 577–585.

Mohamed Elaraby and Muhammad Abdul-Mageed. 2018. Deep models for arabic dialect identification on benchmarked data. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 263–274.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Benjamin D Horne and Sibel Adali. 2017. This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. *arXiv preprint arXiv:1703.09398*.

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.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

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

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *AAAI*, volume 333, pages 2267–2273.

Nut Limsopatham and Nigel Collier. 2016. Learning orthographic features in bi-directional lstm for biomedical named entity recognition. In *Proceedings of the Fifth Workshop on Building and Evaluating Resources for Biomedical Text Mining (BioTxtM2016)*, pages 10–19.

Ferran Pla and Lluís-F Hurtado. 2014. Political tendency identification in twitter using sentiment analysis techniques. In *Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical Papers*, pages 183–192.

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. 2017. A stylometric inquiry into hyperpartisan and fake news. *arXiv preprint arXiv:1702.05638*.

Daniel Preoţiuc-Pietro, Ye Liu, Daniel Hopkins, and Lyle Ungar. 2017. Beyond binary labels: political ideology prediction of twitter users. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 729–740.

Xiaoyan Qiu, Diego FM Oliveira, Alireza Sahami Shirazi, Alessandro Flammini, and Filippo Menczer. 2017. Limited individual attention and online virality of low-quality information. *Nature Human Behavior*, 1:0132.

Arun Rajendran, Chiyu Zhang, and Muhammad Abdul-Mageed. 2019. Happy together: Learning and understanding appraisal from natural language. In *Proceedings of the AAAI2019 Second Affective Content Workshop (AffCon 2019)*, pages 00–00.

Greg Sandoval. 2008. Whos to blame for spreading phony jobs story? *CNet News*, pages 4–46.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. Classifying relations via long short term memory networks along shortest dependency paths. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1785–1794.

Chunting Zhou, Chonglin Sun, Zhiyuan Liu, and Francis Lau. 2015. A c-lstm neural network for text classification. *arXiv preprint arXiv:1511.08630*.

Peng Zhou, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, and Bo Xu. 2016a. Text classification improved by integrating bidirectional lstm with two-dimensional max pooling. *arXiv preprint arXiv:1611.06639*.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016b. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 207–212.