Brenda Starr at SemEval-2019 Task 4: Hyperpartisan News Detection

Olga Papadopoulou, Giorgos Kordopatis-Zilos, Markos Zampoglou,
Symeon Papadopoulos, Yiannis Kompatsiaris
Centre for Research and Technology Hellas, Information Technologies Institute,
Thessaloniki, Greece
(olgapapa,georgekordopatis,markzampoglou,papadop,ikom)@iti.gr

Abstract

In the effort to tackle the challenge of Hyperpartisan News Detection, i.e., the task of deciding whether a news article is biased towards one party, faction, cause, or person, we experimented with two systems: i) a standard supervised learning approach using superficial text and bag-of-words features from the article title and body, and ii) a deep learning system comprising a four-layer convolutional neural network and max-pooling layers after the embedding layer, feeding the consolidated features to a bi-directional recurrent neural network. We achieved an F-score of 0.712 with our best approach, which corresponds to the mid-range of performance levels in the leaderboard.

1 Introduction

The emerging issue of online disinformation has lately attracted the public attention and is perceived as a major risk for democracy and society. Media content (text, images, videos) is often disseminated on the Internet with the purpose of manipulating public opinion. Hyperpartisan news detection is a problem arising as a result of the intention of publishers to influence readers in favour of a given party, idea or person. The SemEval 2019 Task 4 (Kiesel et al., 2019) seeks solutions to this challenge, in particular text-based approaches that can detect hyperpartisan news articles.

We experimented with two approaches: i) a standard supervised learning approach using superficial text and bag-of-words features, and ii) a deep learning system. We deployed the developed systems on TIRA (Potthast et al., 2019) (a platform that supports software submissions) and its evaluation was conducted on unseen news articles. The results of our submissions, which are presented in Table 1, are promising, yet there is still considerable room for improvement. Our best resulting approach was the deep learning system, which scored an F-score of 0.712. The implemented approaches are described below along with additional experiments that were conducted on the provided training and validation datasets.

2 Data

The dataset provided by the organizers of the task (Kiesel et al., 2019) consists of news articles, half of which are labelled as hyperpartisan. It is split into two sets, the training and the validation set, where for each article the article title, body and published date are provided. The training set consists of 500.000 news articles and it is used as training set for the presented experiments and the provided validation set (150.000 news articles) is used for validating the approaches. A small dataset of 645 news articles, manually annotated, is also provided but not used in the following experiments neither as training nor as validation data. For the evaluation phase, two small datasets of 628 and 4000 articles are provided. The first, called by-article test dataset, is labeled through crowdsourcing on an article basis while the latter, named by-publisher test dataset, is labeled by the overall bias of the publisher as provided by BuzzFeed journalists and MediaBiasFactCheck.com.

A pre-processing step is applied on both the article title and body in order to clean the text and prepare it for the subsequent machine learning steps. The Natural Language Toolkit (NLTK) (Bird et al., 2009) was used to implement this step. First, the text is split into sentences and then each sentence is split in tokens. Lemmatization is applied on each token in order to group together the inflected forms of a word and subsequently remove the stop words based on a list of commonly agreed stop words provided by the NLTK.

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 924–928
Minneapolis, Minnesota, USA, June 6–7, 2019. ©2019 Association for Computational Linguistics

	By article test set			By publisher test set		
	Precision	Recall	F-score	Precision	Recall	F-score
SuCla	0.556	0.643	0.596	0.535	0.809	0.644
BOW	0.542	**0.971**	0.696	**0.627**	0.808	0.706
DL	**0.592**	0.895	**0.712**	0.608	**0.860**	**0.712**

Table 1: Evaluation results on the two unseen test sets provided by SemEval-2019 Task 4.

3 Proposed Approach

We experimented with three approaches:

- **SuCla:** a simple classifier based on *superficial* features extracted from the article text (e.g. *number of words*, *contains pronouns*, *number of explanation marks*) and building supervised machine learning models;
- **BOW:** a 'bag-of-words' text classifier;
- **DL:** a deep learning system based on convolutional neural networks (CNN) (LeCun et al., 2015) and recurrent neural networks (RNN) (Medsker and Jain, 1999).

These are further detailed in the next sections.

In the experiments reported here, the training set was used for building the models and the validation set for calculating the evaluation measures: precision, recall and F-score[1]. The decision threshold is set to 0.5 where probabilities $\geq$ 0.5 indicate hyperpartisan articles and $<$ 0.5 non hyperpartisan. Regarding the submissions to the task through the TIRA platform, training was conducted offline by concatenating the training and validation sets as input and then, the trained models were deployed to TIRA to classify the new, unseen news articles.

3.1 Superficial Features Classifier (SuCla)

This simple approach is an adaptation of the one introduced in (Boididou et al., 2018), which was used to assess the credibility of Twitter posts. We extracted a set of superficial features from the article title, which are a subset of the *tweet-based* features presented in (Boididou et al., 2018). These are listed in Table 2. In (Boididou et al., 2018), further information about the Twitter user who posted the tweet was used, but such information is not available for the article publisher in this task.

#	Title-based features
01	Text length
02	Number of words
03	Contains question mark (Boolean)
04	Contains exclamation mark (Boolean)
05	Contains 1st person pronoun (Boolean)
06	Contains 2nd person pronoun (Boolean)
07	Contains 3rd person pronoun (Boolean)
08	Number of uppercase characters
09	Number of positive sentiment words
10	Number of negative sentiment words
11	Number of slang words
12	Has : symbol (Boolean)
13	Number of question marks
14	Number of exclamation marks
15	Number of nouns

Table 2: List of features extracted from the article title.

We extracted the title-based features on the training and validation sets. The extracted 15-dimensional feature vectors were first normalized in the [0,1] range and then fed to a Radial Basis Function (RBF) kernel SVM. The model parameters were calculated using a grid searching method. The software was deployed to TIRA and evaluated on the unseen articles of the test set. The normalization of test article features was conducted using the scaling parameters computed from the training set. Then, articles were classified as hyperpartisan or not with a score in the [0,1] range: the higher the score the more likely the article is hyperpartisan. The precision, recall and F-measure of this run are presented in Table 1 for the two test sets of unseen articles (by-pyblisher and by-article). The resulting F-scores of 0.596 and 0.644 for the by-article and by-publisher test set respectively indicate that this approach performs better than random but requires more distinctive features to further improve the accuracy.

3.2 Bag-of-words Classifier (BOW)

A text item, in our case the article title or body, can be represented as a vector of word occurrences. This is the well-known and widely used 'bag-of-words' (BOW) model. For building the BOW, we

[1] https://en.wikipedia.org/wiki/Precision_and_recall

	Precision		Recall		F-measure	
	Title	Body	Title	Body	Title	Body
MNB	0.54	0.54	0.66	0.79	0.59	0.65
RF	0.56	0.54	0.74	0.68	0.64	0.60
LR	0.58	0.56	0.79	0.81	**0.67**	0.66

Table 3: Evaluation results for Bag of Words on article title and body. Three classifiers are evaluated: Multinomial Naive Bayes (MNB), Random Forest (RF) and Logistic Regression (LR).

started with the clean text resulting from the preprocessing step described in Section 2 and counted the number of occurrences of each word from two vocabularies that were created based on the training set, and had a size of 64,663 and 364,359 words for the title and the body respectively. Three classifiers were evaluated: a) Multinomial Naive Bayes (MNB), b) Random Forest (RF) and c) Logistic Regression (LR). The obtained test results are presented in Table 3. According to it, LR outperforms the other two, irrespective of whether the article title or body is used as input. The resulting F-scores are 0.67 (title) and 0.66 (body). The BOW counts the number of times a word appear in the text of an article (term frequency) regardless of its appearance in other articles. In addition, we applied the Term Frequency-Inverse Document Frequency (TF-IDF), which adapts the term weight in relation to the times that this term appears in all articles. However, the resulting F-score of 0.58 (title) and 0.66 (body) for LR indicated that classification performance would suffer. Additionally, in the attempt to take advantage of both the title and body text, we implemented a fusion step based on averaging the prediction scores of the individual models. As a result, a minor increase of the F-score to 0.69 was obtained at the expense of additional complexity.

The LR classifier was finally trained on the full set of articles (both training and validation sets) and article title. The new BOW model was deployed to TIRA to classify the unseen news articles. This led to slightly better results as presented in Table 1. Compared to the SuCla approach, the BOW performance is significantly better, especially on the by-article dataset.

3.3 Deep Learning System (DL)

An overview of the employed network architecture, which was devised for the task, is presented in Figure 1.

The input to the network is the vectorized form of the articles' title and body. The input text is pre-processed as described in Section 2. An additional step is applied in order to form the text so that the inputs to the network have the same shape for each article. More specifically, for each article we retain the first 64 sentences, and for each sentence the first 64 words. This results in a (64x64)-dimensional tensor that is provided as input to the network. Zero padding is applied in order to fill missing words and/or sentences.

The input of the network is provided to an Embedding layer, to map each word of the input text to a *word embedding*. We used the pre-trained FastText word embeddings (Mikolov et al., 2018) of size 300. The weights of this layer are not updated during learning. In that way, we overcome the limitation of a bounded vocabulary, imposed by the training set, and the network can process words outside the training sets since they exist in the vocabulary of FastText. The output of this layer is a tensor of (64x64x300) for each article.

Then, we apply multiple convolution filters with different kernel sizes on the output of the Embedding layer. In that way, the network can capture word sequence structure in different granularity levels. The convolutional layers are used with kernel sizes of (1x1), (1x3), (1x5), and (1x7) and in combination with a ReLU activation function. The output of each convolutional layer is a (64x64x128)-dimensional tensor. The outputs of the four convolutional layers are then concatenated on the channel axis (the last tensor dimension) to form a (64x64x512)-dimensional tensor per article. Finally max-pooling is performed over the word axis, i.e., the maximum value per channel and sentence is extracted. To this end, the Embedding and Convolutional layers of the network capture word-level information from the article text.

After max-pooling on the outputs of the Convolutional layers, the (64x512)-dimensional tensors are given to a bidirectional Recurrent Neural Network (bi-RNN) (Schuster and Paliwal, 1997) that

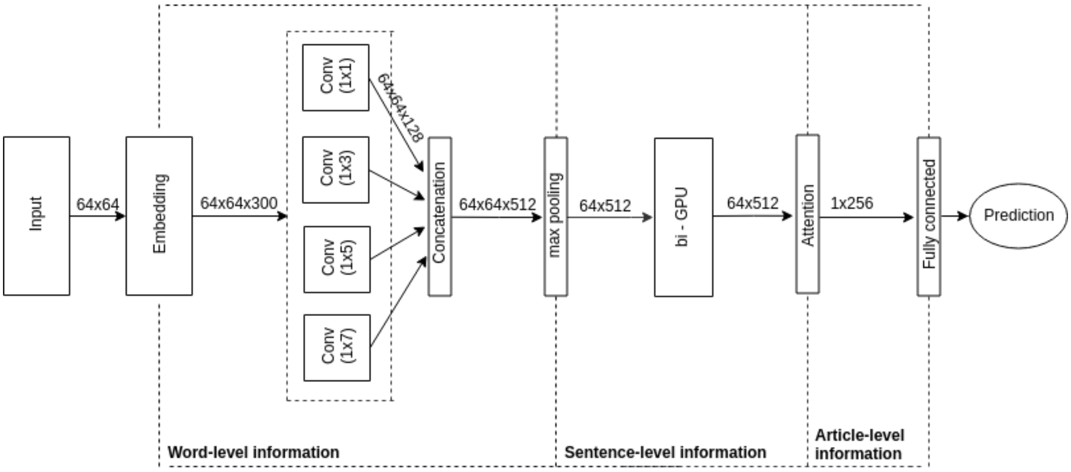

Figure 1: The deep learning architecture developed for classifying a news item as hyperpartisan or not.

calculates sentence vectors by taking into account the neighbor sentences. More precisely, for every article sentence i, the hidden vector h_i summarizes the neighbor sentences around sentence i but still focuses on that sentence. We employed the bidirectional Gated Recurrent Units (bi-GRU) (Cho et al., 2014) as the recurrent unit of the bi-RNN, which is an improved version of the standard recurrent unit. The output of the bi-GRU layer is provided to an attention mechanism (Yang et al., 2016) that weights each sentence vectors based on their similarity to a sentence-level context vector, and then averages the weighted vectors to single vector. The result of the Attention layer is a (1x256)-dimensional vector.

At the final stage, the network captures article-level information. The output of the Attention layer is fed to a fully connected layer to get the final prediction of the network. In this layer, we apply Sigmoid activation to map the output to the [0,1] range, which represents the probability of the article being hyperpartisan. Finally, the network is trained with the binary cross-entropy loss function, weight decay with a $5 * 10^{-4}$ regularization factor, Adam (Kingma and Ba, 2014) optimizer, and 10^{-3} learning rate. Training is done for 100 epochs with a batch size of 32 articles, and the best network is selected based on the performance on the validation set.

This method performs better than the other two approaches, achieving an F-score of 0.712 (Table 1) for both test sets.

3.4 Ideal Fusion

We implemented an ideal fusion method in order to examine the complementarity between the three proposed approaches. This is a theoretical scheme (oracle) which takes the outputs of the individual approaches and selects the correct classifier: at least one model needs to classify correctly an article. An F-score of 0.85 is achieved on the validation set, far better than the individual classifiers accuracy (SuCla: 0.51, BOW: 0.67, DL: 0.65) indicating that the models bring complementary information, which make them good components of a combined model.

4 Conclusions

This paper summarized our participation in SemEval-2019 Task 4, where we aimed at the challenge of Hyperpartisan News Detection. We tried to approach the problem from the perspective of standard supervised learning techniques, as well as more complex deep learning approaches. While none of the methods gave groundbreaking results, our set of experiments and observations provides a solid basis for future research on the problem. In particular, we intend to conduct more extensive analysis on the annotated data and extract patterns that will be more representative and distinctive for the problem at hand. Moreover, we will consider combining the three proposed approaches with the aim of creating a stronger and more accurate combined model. The significant increase in performance of the ideal fusion method points out the benefits of such a strategy.

5 Acknowledgments

This work is supported by the WeVerify project, which is funded by the European Commission under contract number 825297.

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

Christina Boididou, Symeon Papadopoulos, Markos Zampoglou, Lazaros Apostolidis, Olga Papadopoulou, and Yiannis Kompatsiaris. 2018. Detection and visualization of misleading content on twitter. *International Journal of Multimedia Information Retrieval*, 7(1):71–86.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

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.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature*, 521(7553):436.

Larry Medsker and Lakhmi C Jain. 1999. *Recurrent neural networks: design and applications*. CRC press.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

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.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.