Cardiff University at SemEval-2019 Task 4: Linguistic Features for Hyperpartisan News Detection

Carla Pérez-Almendros, Luis Espinosa-Anke and Steven Schockaert
School of Computer Science and Informatics
Cardiff University, UK
{perezalmendrosc,espinosa-ankel,schockaerts1}@cardiff.ac.uk

Abstract

This paper summarizes our contribution to the Hyperpartisan News Detection task in SemEval 2019. We experiment with two different approaches: 1) an SVM classifier based on word vector averages and hand-crafted linguistic features, and 2) a BiLSTM-based neural text classifier trained on a filtered training set. Surprisingly, despite their different nature, both approaches achieve an accuracy of 0.74. The main focus of this paper is to further analyze the remarkable fact that a simple feature-based approach can perform on par with modern neural classifiers. We also highlight the effectiveness of our filtering strategy for training the neural network on a large but noisy training set.

1 Introduction

In the era of misinformation, the challenge of differentiating reality from frames, facts from opinions, is becoming increasingly important. Concepts such as *Fake News*, *Fact Checking* or *Post-Truth Era*, generally unknown a few years ago (Lewandowsky et al., 2017), started to play an important part in media, academic papers and even in Natural Language Processing (NLP) tasks (Rashkin et al., 2017; Shu et al., 2017; Wang, 2017). Nowadays, strongly opinionated news stories can offer biased information, as is the case with hyperpartisan articles. A text is considered hyperpartisan when it is highly polarized towards an extreme position. Potthast et al. (2018) analyzed hyperpartisanism in relation to fake news, to discover that a very similar writing style could be associated both with right-wing and left-wing polarized stories. This shared style of biased articles was different from that of mainstream articles. Task 4 in SemEval 2019 (Kiesel et al., 2019) consisted in a classification challenge where news articles had to be sorted out as hyperpartisan or non-hyperpartisan. Our approach addressed this challenge via two different models, which included 1) an SVM classifier based on word embeddings averages and handcrafted linguistic features and 2) a recurrent BiLSTM neural network classifier trained on filtered data. The reason and process for filtering data will be explained in section 3.2.

Word embeddings have remained central to the state-of-the-art approaches in NLP since the introduction of Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) models. In addition to modelling word meaning, embeddings can also be applied to longer units of significance, such as phrases, sentences, paragraphs or entire documents (Le and Mikolov, 2014). However, although such vector representations often enable the best results in a variety of NLP challenges, some tasks still benefit from linguistically inspired approaches. In fact, recent work has proved how linguistic features and stylometry could improve the performance of deep learning techniques. The already mentioned work in Hyperpartisan and Fake News detection (Potthast et al., 2018) applied stylometry based on linguistic features to identify strongly biased articles. Bag of words, stop words, part of speech and readability scores are some of the features analyzed by the authors. They also focus on quotes, measuring their length and counting their appearances in a text. Rhetorical questions or the appearance of personal pronouns, among many others linguistic features, also helped to classify suspicious vs trusted news posts on Twitter (Volkova et al., 2017). The number of adverbs or swear words has also been used for fact checking purposes (Rashkin et al., 2017). Inspired by these previous works and their results, these linguistic features are some of the ones we apply in our first model.

The second model is based on word embeddings as input for a neural network. Specifically, we used a Convolutional Neural Network (CNN) combined with a Bidirectional Long-Short Term Memory (BiLSTM) network. The main novelty of this approach lies in the preprocessing step which filters the training data. This strategy is used because the bulk of the training data only provides a weak supervision signal, which we found too noisy to use directly.

2 Data

The data provided for this task consisted of a training set of 645 news articles, manually labeled as hyperpartisan or non-hyperpartisan. In addition to this gold standard data, a larger dataset of 600,000 training articles and 150,000 validation articles was provided. These complementary documents were labeled automatically depending on their source media. If the media was considered hyperpartisan, the article was labeled as such, but without analyzing its content. These automatically tagged articles also included further labels (referring to the publisher rather than the article itself), which have not been used in our system. Our first model exclusively relied on the manually labeled articles (645) for training, while the second model also took advantage of the larger set of weakly labeled articles (750000).

3 Our Approach

The dataset was preprocessed by changing the text to lower case and then applying tokenization. For our first model, sentence tokenization was applied and articles were preprocessed with part-of-speech tagging (PoS), using the NLTK library (Bird et al., 2009). While Potthast et al. (2018) kept and analyzed quotes in texts, we chose to delete them in both datasets. Our first experiments showed that, in a small number of cases, keeping quotes drove our system to misclassification because while an article could quote hyperpartisan statements of others, the document itself did not necessarily have an extreme position towards a topic or event. In both approaches, we only considered the main text of the article, discarding its title.

3.1 Model 1: Combining Document Embeddings with Linguistic Features

Document embeddings were built for each article. For doing so, we first computed embeddings for all sentences by averaging the pre-trained GloVe vectors (Pennington et al., 2014) for all the words occurring in them. To this end, we used the uncased Common Crawl pretrained GloVe embeddings, with 300 dimensions and a vocabulary of 1.9 million words. The average of all the sentences in an article was then computed to obtain a single vector representation of the news article. To complement this document vector, we identified a number of document-level discriminant linguistic features to classify a text as hyperpartisan or non-hyperpartisan. The selected features are as follow:

- **`excl`**: total number of exclamation marks
- **`quest`**: total number of question marks
- **`adj`**: percentage of tokens which are adjectives
- **`adv`**: percentage of tokens which are adverbs
- **`insults`**: total number of insults or swear words[1]
- **`first_pers`**: total number of times that the first person personal pronoun *I* was used
- **`sent_length`**: average length of sentences
- **`min_sent`**: length of the shortest sentence
- **`max_sent`**: length of the longest sentence

These feature values were concatenated with the document vector to provide the input for a linear SVM classifier.

To experiment with different configurations of our method, we used the 645 manually labeled articles with 5-fold cross-validation. Document embeddings and linguistic features vectors were considered both separately and concatenated as input for different classifiers, namely Random Forest, Logistic Regression and Support Vector Machine (SVM), with different parameters. In all cases, we used the implementations from the Scikit-Learn Machine Learning Library (Pedregosa et al., 2011). After testing the different options, a concatenated vector of document embeddings and linguistic features as input for an SVM proved to ob-

[1] A file containing swear words and insults was provided as input for a swear words count function. The file was created with a list of such words extracted from https://www.digitalspy.com/tv/a809925/ofcom-swear-words-ranking-in-order-of-offensiveness/, and then augmented with 2,500 similar insults coming from their *word2vec* nearest neighbours.

tain the best results. We finally trained our combined model as a linear SVM on the entire set of 645 gold standard documents.

3.2 Model 2: Neural Text Classification

Neural networks need large amounts of data to be able to learn. The small dataset of 645 manually labeled articles was clearly too small to train a competitive neural network model. However, we noticed that the large training set of 750K documents, which was labeled based on the publisher, was too noisy, yielding a performance which was far worse than that of the first model. We attempted to surmount this issue via a two-step strategy, in which we first trained a classifier on the small training set, which we used to filter the larger but noisy training set. The goal was to automatically extract from the 750K labeled-by-publisher articles only those which were correctly predicted as hyperpartisan or non-hyperpartisan by this initial classifier. The strategy which we found to perform best was the following:

1. Using half of the 645 labeled articles, we trained three classifiers:

   - a linear SVM based on the linguistic feature set described in Section 3.1,
   - a linear SVM based on the document embedding
   - and a standard neural classifier using a convolutional layer followed by a bi-LSTM layer (CBLSTM).

   The two first classifiers were included after they had been tested in our first model, and with the parameters previously explained. The choice of using a combination of CNN and LSTM for our third classifier stems from previous work where either or both architectures combined proved to be useful in document classification (Kim, 2014; Xiao and Cho, 2016). Concerning the choice of hyperparameters, we used 100 convolutional filters of size 5, and one-token strides. The output of the CNN layer was then passed to a max-pooling layer (where pool size was set to 4), and this output was passed to a bidirectional LSTM layer which produces two 100d vector outputs, which after concatenation, were passed to a final 2d softmax layer. We implemented this model using the *keras*[2] library.

2. We then trained a meta-classifier on the remaining half of the 645 articles, which used the predictions of these three individual classifiers as features, and which finally generated a final prediction. For this meta-classifier we used a linear SVM.

3. Once trained, the meta-classifier was applied to the 750K labeled-by-publisher articles. Whenever the ground-truth label agreed with the prediction of our metaclassifier, the article was retained in our filtered dataset.

4. Through this process, we obtained around 150K refined labeled articles that we used for training another CBLSTM, replicating the same process and parameters used in step 1. This last refined model was the one applied to the test set.

4 Analysis

We will focus our analysis on the first model, given that its result is perhaps most surprising. We observed that exclamation and question marks were present in non-hyperpartisan and mainstream articles, but a high occurrence of these features is nonetheless clearly correlated with hyperpartisanism. Adjectives and adverbs tended to be more frequent in hyperpartisan texts as well. Insults or swear words were extremely scarce in non-hyperpartisan articles, so their presence is a strong indicator for hyperpartisanism. The personal pronoun *I* denotes a personal text, and for this reason is more common in strongly opinionated articles. Average sentence length was not found to be particularly informative. On the other hand, we found that the shortest sentences in hyperpartisan articles tend to be shorter than those in non-hyperpartisan articles. In a similar way, the longest sentences were also slightly longer in extreme news stories. These results have been summarized in Table 1.

Further experiments to assess the discriminatory power of each linguistic feature were performed, although these took place after the submission for this SemEval task. A 5-fold cross validation on the 645 gold-standard articles dataset was applied to estimate the performance of the model in each case. As can be seen in Table 2, the linguistic features on their own are sufficient for achieving an accuracy of 66%. Combining them with document embeddings, the results reached 73.2%. However, a deeper analysis showed that

[2] `https://keras.io/`

	Hyp.	Non Hyp.
excl (avg)	1.30	0.63
quest (avg)	2.43	1.20
adj (avg)	9.00	8.40
adv (avg)	4.10	3.20
insults (avg)	0.07	0.01
first_pers (avg)	3.26	1.78
sent_length (avg)	22.47	24.43
min_sent(median)	2.00	4.00
max_sent(median)	52.00	47.00

Table 1: Linguistic features extracted from 645 articles dataset.

some linguistic features were actually deteriorating the general performance of the system. For instance, the linguistic features model alone performed better when `excl` were not accounted for. On the other hand, `insults`, `adj` and `adv`, respectively, were the most discriminant features, leading to the biggest drop in performance when discarded. Here, we would like to highlight that, surprisingly, the feature `adv` reduces the performance of the combined model, where eliminating it allows the system to reach 75% of accuracy. The feature `first_person` is also reducing the score of the combined system. However, whenever we omitted two or more linguistic features, the performance of the combined model dropped below 72.7%, which is the accuracy achieved by the document vectors on their own. Therefore it seems safe to conclude that our linguistic features share some information that, combined, provide complementary evidence for document embeddings.

	Ling. Feat.	Comb. model
complete model	**.660**	**.732**
- excl	.666	.738
- quest	.663	.730
- adj	.642	.730
- adv	.651	.750
- insults	.640	.738
- first_pers	.662	.742
- sent_length	.660	.735
- min_sent	.662	.730
- max_sent	.662	.736
only document embeddings	-	**.727**

Table 2: Ablation results for the first model in terms of accuracy.

5 Results and Discussion

Both our systems obtained around 74% accuracy in SemEval 2019 task 4: Hyperpartisan News Detection under the team name of Ankh-Morpork Times (Potthast et al., 2019). This constitutes an improvement of 28 percentage points over the provided baseline. In the SemEval competition, our team got the 16th position out of 42 participants. Our main contribution was to show that a simple approach, based on document embeddings and linguistic features, can obtain the same accuracy as a typical neural text classifier.

Overall, there are several lessons learned from our participation in this task, which we will try to develop in future work. For example, we confirmed that, although the community tends to rely on the performance of word vectors, linguistic features can complement word vector based representations in a meaningful way in text classification. In addition, further analysis in our work showed that we could have improved our performance with a better selection of linguistic features. Therefore, for future work, we aim at providing a more reliable model which takes into account more complex linguistic features. For instance, we believe that looking at sentences' modality and sentiment, as well as assessing the polarity of adjectives and adverbs in a text, should give valuable extra information for the task.

As a secondary contribution, we also proposed a technique for filtering noisy data. It is known that neural networks perform well for large training sets, but sometimes a large accurately labeled dataset cannot be obtained. To this end, we created a meta-classifier trained on a smaller gold standard dataset and applied to larger, noisy data for obtaining a filtered higher-quality training set.

6 Namesake

Ankh-Morpork is the biggest city in the Discworld, the fictional world that gives name to the famous fantasy book series by Sir Terry Pratchett. And Ankh-Morpork Times is its first, biggest and most famous newspaper, and covers in a peculiar and surreal way the no less surreal events happening in this flat world. And sometimes, we must admit, with quite a hyperpartisan point of view.

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.".

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*.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196.

Stephan Lewandowsky, Ullrich KH Ecker, and John Cook. 2017. Beyond misinformation: Understanding and coping with the post-truth era. *Journal of Applied Research in Memory and Cognition*, 6(4):353–369.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher 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.

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.

Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2931–2937.

Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 19(1):22–36.

Svitlana Volkova, Kyle Shaffer, Jin Yea Jang, and Nathan Hodas. 2017. Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on twitter. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 647–653.

William Yang Wang. 2017. " liar, liar pants on fire": A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648*.

Yijun Xiao and Kyunghyun Cho. 2016. Efficient character-level document classification by combining convolution and recurrent layers. *arXiv preprint arXiv:1602.00367*.