Dick-Preston and Morbo at SemEval-2019 Task 4: Transfer Learning for Hyperpartisan News Detection

Tim Isbister
Swedish Defence Research Agency
Stockholm, Sweden
tim.isbister@foi.se

Fredrik Johansson
Swedish Defence Research Agency
Stockholm, Sweden
fredrik.johansson@foi.se

Abstract

In a world of information operations, influence campaigns, and fake news, classification of news articles as following hyperpartisan argumentation or not is becoming increasingly important. We present a deep learning-based approach in which a pre-trained language model has been fine-tuned on domain-specific data and used for classification of news articles, as part of the SemEval-2019 task on hyperpartisan news detection. The suggested approach yields accuracy and F1-scores around 0.8 which places the best performing classifier among the top-5 systems in the competition.

1 Introduction

In today's polarized media and political landscapes, the challenge of determining whether a news article is biased or not is highly topical. In the hyperpartisan news detection task (Kiesel et al., 2019) of the International Workshop on Semantic Evaluation (SemEval) 2019, the task is to predict whether a given news article text follows a hyperpartisan (extreme one-sided) argumentation or not, i.e., whether it exhibits blind or prejudiced allegiance to one party, cause, or person (Potthast et al., 2019). As part of this challenge, participating research teams got access to two datasets:

1. **by-publisher**: A well-balanced dataset consisting of 750,000 articles in which the data have been labeled by the overall bias of the *publisher*, as provided by journalists or fact-checking sites.

2. **by-article**: A smaller dataset consisting of 645 articles for which crowdsourcing workers have agreed on the labeling of the *articles* as being hyperpartisan (37%) or not (63%). A similar but more well-balanced test dataset (to which the participating teams have not got direct access) has been used for evaluating the accuracy, precision, recall, and F1-score of systems developed by the participating research teams.

In this system description paper we present the results for the two participating research teams from the Swedish Defence Research Agency (FOI): 1) **dick-preston** and 2) **morbo**.

The teams contributed with separate systems for the early-bird deadline and for the final submission. In the early phase we used traditional machine learning classifiers such as logistic regression and support vector machines (SVMs), built upon traditional text features such as word and character n-gram term frequencies (weighted with inverse document frequency). These classifiers have been used as baselines to which more "modern" NLP classifiers have been compared. For the final submission both teams made use of transfer learning-based Universal Language Model Fine-Tuning (ULMFiT) models. The difference in the teams' final systems is the percentage of data used for training/validation splits when fine-tuning the models and the number of epochs for which the models were trained. Despite that only a few hundred examples were used for fine-tuning the pre-trained ULMFiT-models, accuracies and F1-scores of approximately 0.8 were achieved on the unseen test data. This resulted in a fifth place for the team **dick-preston** and seventh place for the team **morbo** out of 42 participating teams, as reported on the competition leaderboard[1].

The rest of this paper is structured as follows. In Section 2, we present the machine learning algorithms and features which have been used for building the hyperpartisan news article classifiers used in the competition. In Section 3 we outline

[1] https://pan.webis.de/semeval19/semeval19-web/leaderboard.html

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 939–943

the conducted experiments, present the used hyperparameters, and describe the obtained results. Finally, we present overall conclusions and discuss ideas for future work in Section 4.

2 Method

In the early phase of the competition, both FOI teams experimented with traditional machine learning algorithms such as Naïve Bayes, logistic regression, and support vector machines (SVMs), taking sparse text features such as word and character n-grams as input. These methods have been used as baselines to which more novel algorithms have been compared. The FOI baseline methods are briefly presented in Section 2.1.

For the final system submission we have used more "modern" NLP methods. More specifically, Universal Language Model Fine-Tuning (ULMFiT) was utilized. ULMFiT is a natural language processing (NLP) transfer learning algorithm introduced in (Howard and Ruder, 2018). ULMFiT is one of several language model-based transfer learning algorithms developed in 2018 which have been shown to yield state-of-the-art results on several NLP tasks. Approaches such as ELMo (Peters et al., 2018), OpenAI GPT (Radford et al.), and BERT (Devlin et al., 2018) have arguably received more attention than ULMFiT, but we selected to implement our final systems using ULMFiT due to its straightforward implementation in the fastai library[2], and its promising results also on small datasets (Howard and Ruder, 2018). ULMFiT is presented in more detail in Section 2.2.

2.1 Baseline Classifiers

As baseline classifiers we have made use of traditional "shallow" machine learning algorithms like logistic regression, SVMs, etc. An extensive list of the tested algorithms can be found in the experiment descriptions in Section 3. A detailed explanation of such classifiers is outside the scope of this paper but we refer the interested reader to (Hastie et al., 2001) for an excellent introduction to such approaches.

As input features to our baseline classifiers we have used term frequencies of n-grams. In the most basic case of 1-grams (unigrams), this means that for each token in the dataset (tested on character as well as word level) we count the number of times the specfic token (e.g., the word "Trump") appears. In the case of 2-grams (bigrams) we do the same, but then for pairs of tokens (e.g., "President Trump"). To account for tokens which appear frequently in all kinds of news articles (thereby making them less valuable for prediction of the target class) we weigh the term frequencies by their inverse document frequency. Various strategies such as only including the most frequently occuring tokens have also been utilized. Details of which strategies that have been tested in our experiments are given in Section 3.

2.2 ULMFiT

As the basis of our ULMFiT models we have used a pre-trained language model trained on the English Wikitext-103 (Merity et al., 2016), which in total consists of more than 28,000 preprocessed Wikipedia articles and over 100 million words. The pre-trained language model consists of a word embedding layer connected to a three-layered unidirectional left-to-right AWD-LSTM (Merity et al., 2017). The AWD-LSTM utilizes several regularizations strategies such as a DropConnect mask on the hidden-to-hidden recurrent weights and variable length backpropagation through time (BPTT) sequences. Given a sequence of N tokens, a left-to-right language model can be used to compute the probability of the sequence of tokens:

$$P(t_1, t_2, \ldots, t_N) = \prod_{k=1}^{N} P(t_k | t_1, t_2, \ldots, t_{k-1}) \quad (1)$$

Language models are powerful in that they can "teach themselves" a lot about language by simply letting them iteratively predict the next word in a sequence on large amounts of (otherwise unlabeled) training data. Throughout this process, the parameters in the ULMFiT AWD-LSTM layers implicity learn about both syntax and semantics as these are helpful for predicting the next word in a sequence.

In next step, the pre-trained language model has been fine-tuned on the 645 articles in the manually crowdsourced **by-article** dataset. The reason for this fine-tuning is that the news articles most likely stem from a different data distribution, compared to the Wikipedia articles on which the language model originally have been trained. During the language model fine-tuning, discriminative learning and slanted triangulated learning rates (SLTR) was used, as outlined in the original

[2] https://github.com/fastai/fastai

ULMFiT paper (Howard and Ruder, 2018). The language model could most likely have been improved upon more by making use of the larger **by-publisher** dataset. However, we were interested in how good the ULMFiT model would perform on a very limited dataset.

In the last step, the fine-tuned language model has been augmented with two linear blocks separated by a rectified linear unit (ReLU) activation function. The last linear block consists of just two output nodes with a softmax activation, giving as output a probability of the current news article being hyperpartisan or not given the fine-tuned model. Regularization in the form of dropout and batch normalization is applied to the linear blocks in order to allow for better generalization. Moreover, gradual unfreezing is used in order to avoid catastrophic forgetting, a phenomenon which previously has been common when trying to fine-tune pre-trained language models.

3 Experiments and Results

For the first part of the competition we experimented with baseline classifiers to which we later on could compare the classification accuracy of more advanced algorithms on hold-out validation datasets constructed from the training data. The experiments with the baseline classifiers were performed using scikit-learn, while latter experiments have been carried out using various deep learning frameworks (including TensorFlow and Keras). The final ULMFiT classifier implementations and experiments have been carried out using PyTorch and the fastai library.

3.1 FOI Baseline Classifier Experiments

We first experimented with a number of simple classifiers which were used as baselines:

- SVM (LinearSVC)
- Logistic Regression
- Random Forest
- Gradient Boosting
- Naïve Bayes
- NBSVM

We used scikit-learn to conduct a grid-search over various hyperparameters for these classifiers in order to find suitable optimized baseline models. In this section we will focus on the hyperparameters of the SVM classifier and its input features as this performed the best among the evaluated classifiers.

A consistent result for all the tested classifiers was that they performed better when creating the n-gram features described in last section from the text context of the news articles rather than only using the shorter titles. As input to the classifier we combined the 1000 word unigrams and bigrams ranked highest in terms of TF-IDF and the 1000 character unigrams and bigrams ranked highest in terms of TF-IDF. For the SVM we used a linear kernel and the regularization parameter C was set to 0.38. Using these parameters we obtained a weighted F1-score of 0.78 when applying stratified 10-fold cross validation on the training data. When the same model was trained on 100 % of the training data and submitted for evaluation on the test data as part of the early-bird deadline we obtained an accuracy of 0.77. This is a rather strong baseline as it would have resulted in a top-10 result in the final leaderboard (if the final FOI classifiers would not have been submitted).

3.2 FOI ULMFiT Classifier Experiments

The embedding layer of our ULMFiT classifiers uses word embeddings with an embedding size of 400. For the sequential linear layers that have been attached to the pre-trained LSTM layers we have used a momentum of 0.1 for the BatchNorm and a dropout probability p of 0.2 for the first linear layer and 0.1 the last linear layer. We have gradually unfreezed different blocks of the model to avoid catastrophic forgetting. Different slanted learning rates and number of training epochs have been used for the different submitted FOI classifiers, but we have in general found learning rates around 0.01 to work well for fine-tuning just the last layer, and then using lower magnitude learning rates when unfreezing earlier layers.

We evaluated the fine-tuned ULMFiT classifiers by splitting the available **by-article** dataset into a training set (85 %) and a validation set (15 %). This was attempted on a few random splits for which we consistently reached accuracies on the validation set over 0.95. In the end we submitted models trained on 85 % and 100 % of the training data but the one trained on 85 % performed the best, probably due to overfitting of the other model (which is natural since it was hard to know how

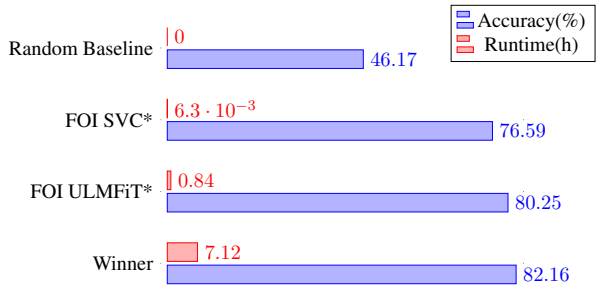

Figure 1: Accuracy and runtime of FOI classifiers in comparison to a random baseline and the winner of the hyperpartisan news detection competition.

many epochs the model should be trained for when not having any separate validation data to evaluate on). When the best performing model was submitted for evaluation on the test set it obtained an accuracy of 0.80 which resulted in a fifth place in the final leaderboard.

In Figure 3.2 we compare the accuracy and running time of our best performing ULMFiT classifier on the test data and contrast them to the corresponding measures for our FOI linear SVM classifier, a random baseline provided by the task organizers, and the classifier developed by the winning team. As can be seen, the accuracy of the ULMFiT classifier is marginally lower than the winning classifier, while the running time seems to be much lower[3].

4 Conclusions

In this paper we have described the ULMFiT classifiers developed by the FOI teams **dick-preston** and **morbo** for the SemEval-2019 challenge of hyperpartisan news article detection. By first fine-tuning a pre-trained language model on the texts and titles of a small dataset consisting of 645 news articles and then fine-tuning two additional linear blocks on humanly annotated labels of these articles, we have achieved accuracy and F1-scores around 0.80 on the task organizers' test dataset. The obtained accuracies resulted in a fifth and seventh place, respectively, out of a total of 42 research teams who submitted their classifiers to the competition. This demonstrates the applicability of novel transfer learning approaches such as ULMFiT to domains for which only very limited amounts of data is available. To the best of our knowledge, this is the first time on which ULMFiT has been attempted on such a small dataset.

4.1 Future Work

The obtained results could have been improved upon by utilizing the larger available **by-publisher** training set for improving the fine-tuning of the language model on the target domain. It is also possible that this larger dataset could have been used for further fine-tuning of the classifier.

Another interesting idea for future research on this dataset would be to train a classifier based on Google AI's BERT, which make use of a deep bidirectional transformer instead of a multi-layered LSTM.

Acknowledgments

This work was supported by the R&D programme of the Swedish Armed Forces.

[3]The submissions were evaluated on a rather slow virtual machine (Potthast et al., 2018) which impact the running times.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2001. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA.

Jeremy Howard and Sebastian Ruder. 2018. Fine-tuned language models for text classification. *CoRR*, abs/1801.06146.

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.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2017. Regularizing and optimizing LSTM language models. *CoRR*, abs/1708.02182.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *CoRR*, abs/1609.07843.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *CoRR*, abs/1802.05365.

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.

Alec Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. Available: https://blog.openai.com/language-unsupervised/.