Adversarial Domain Adaptation for Stance Detection

Brian Xu, Mitra Mohtarami, James Glass
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA, USA
`{bwxu, mitram, glass}@mit.edu`

Abstract

This paper studies the problem of stance detection which aims to predict the perspective (or stance) of a given document with respect to a given claim. Stance detection is a major component of automated fact checking. As annotating stances in different domains is a tedious and costly task, automatic methods based on machine learning are viable alternatives. In this paper, we focus on adversarial domain adaptation for stance detection where we assume there exists sufficient labeled data in the source domain and limited labeled data in the target domain. Extensive experiments on publicly available datasets show the effectiveness of our domain adaption model in transferring knowledge for accurate stance detection across domains.

1 Introduction

With the rise of social media and microblogs, there has been an increasing awareness of the negative influence of fake news and how it can unfairly influence public opinion on various events and policies [8, 9, 20]. In order to counteract these effects, various organizations are now performing manual fact checking on suspicious claims. However, manual fact checking can't feasibly keep up with the sheer volume of fake claims. A fact-checking process for a given claim is a challenging multi-step process, that typically involves the following steps [19]: (*i*) retrieving potentially relevant documents as evidence for the claim [10, 6], (*ii*) predicting the stance of each document with respect to the claim [14, 2], (*iii*) estimating the trustworthiness of the documents (e.g. in the Web context, the site of a Web document could be an important indicator of its trustworthiness), and finally (*iv*) making a decision based on the aggregation of (*ii*) and (*iii*) for all documents from (*i*) [10].

In this paper, we focus on the second step of the fact-checking process which is the *stance detection* task. Stance detection aims to automatically determine the perspective (or stance) of a document to a claim as *agree*, *disagree*, *discuss*, or *unrelated*. Since there is not enough data with annotated stance labels for many domains, machine learning algorithms can easily produce poor to mediocre performance. One potential approach to alleviate the lack of labeled data is to supplement the data sources with data from other domains. However, it is not a straightforward process as each data source generally has its own unique distributions and nuances. In this paper, we tackle this problem using a transfer learning technique, namely adversarial domain adaptation, to effectively use labeled data from a source domain to improve stance detection performance on a target domain which has limited data. Our contributions can be summarized as follows:

- We are the first to apply adversarial domain adaptation to the problem of stance detection across different sources.
- Our model outperforms the best baseline on a publicly-available benchmark dataset, Fake New Challenge [1].

[1] Available at `www.fakenewschallenge.org`

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

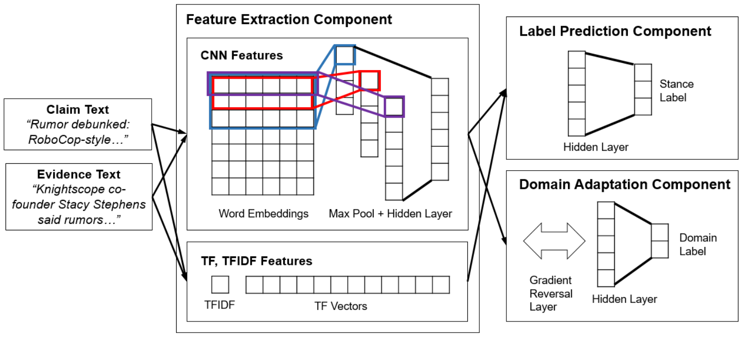

Figure 1: The architecture of our model with domain adaptation component for stance detection.

2 Method

Previously-proposed approaches for stance detection generally contain two components [1, 5, 16]: a feature extraction component followed by a class label prediction component. In this paper, we present a model for stance detection that augments the traditional models with a third component: a domain adaptation component. Our domain adaptation component uses adversarial learning [4] to encourage the feature extraction component to select common–rather than domain-specific–features when input data is from multiple different domains. This allows the model to better leverage *source* domain data for better prediction on data from the *target* domain. The general architecture of our model is shown in Figure 1. As illustrated, the inputs are first given to the "*Feature Extraction Component*" to compute their features and representations. These features are then passed to the "*Label Prediction Component*" and then to the "*Domain Adaptation Component*." In the model, while the both latter components try to minimize their own losses, the feature extraction component attempts to maximize a domain classification loss to encourage better mixture of examples from different domains. The components of the model are described in detail below.

Feature Extraction Component: This component takes the input claim c and document d and converts them to their semantic representations and features. To do this, in this component, we use bag-of-words (BOW)—e.g., TF and TF.IDF-weighted features—and cosine similarity between c and d features. We select these BOW features as these features are useful to filter the documents with 'unrelated' stance labels as shown in section 3.5. Furthermore, we also use a Convolutional Neural Network (CNN) approach [7] for learning representations of claims and documents. We use a CNN because it can capture n-grams and long-range dependencies [21], and can extract discriminative word sequences that are common in the training instances [17]. These traits make CNNs useful for dealing with long documents [15].

Label Prediction Component: This component uses a Multi-Layer Perceptron (MLP) with a fully-connected hidden layer followed by a *softmax* layer which employs *cross entropy* loss as the cost function. This component will predict stance labels as *agree*, *disagree*, *discuss*, or *unrelated* for a given set of claim and document features.

Domain Adaptation Component: We introduce a domain classifier which includes a MLP followed by a *softmax* layer. Given a set of features for a claim-document pair, the domain classifier predicts which domain the features originated from. The domain classifier is an adversary because the model—specifically, the feature extraction component—attempts to maximize the domain classifier loss, while the domain classifier attempts to minimize it. This is because a high domain classifier loss implies that the domain classifier is unable to accurately discern whether a set of features belongs to a *source* or *target* domain. This implies that the features extracted from the input examples are common to both the *source* and *target* domains as we desire. To achieve this adversarial effect, the features from the feature extraction component are passed to a *gradient reversal* layer before being passed to the domain classifier. The gradient reversal layer is a simple identity transform during forward propagation and multiplies the gradient by a negative constant (the gradient reversal constant) during backpropagation [4]. By adding the gradient reversal layer, the desired training behavior can be achieved through normal model training.

3 Experiments and Evaluations

3.1 Datasets

We use the Fake News Challenge (FNC) dataset[1] as *target* data. This data is collected from a variety of sources such as rumour sites, e.g. `snopes.com`, and Twitter accounts such as *@Hoaxalizer*. It contains around 50K claim-document pairs as training data with an imbalanced distribution over stance labels: 73% (*unrelated*), 18% (*discuss*), 7.3%(*agree*), 1.7%(*disagree*). We note that there is a lack of labeled data especially for *agree* and *disagree* stance labels in FNC. Furthermore, we use the Fact Extraction and VERification (FEVER) dataset [18] as *source* data. This dataset is collected from Wikipedia and contains around 145K claim-document pairs as training data with imbalanced distribution over stance labels: 55% (*supported*), 21% (*refuted*), 24%(*Not Enough Information (NEI)*). We discard the examples with NEI labels as there is not any documents assigned to the claims in those examples, and correspond the *supported* and *refuted* labels in FEVER to the *agree* and *disagree* labels in FNC respectively.

3.2 Evaluation Metrics

We use the following evaluation metrics: **Macro-F1**: The average of the F_1 score for each class. **Accuracy**: The number of correctly classified examples divided by their total number of examples. **Weighted-Accuracy**: This metric is presented by FNC [2] which is a two-level scoring scheme. It gives 0.25 weight to the correctly predicted examples as *related* or *unrelated*. It further gives 0.75 weights to the correctly predicted *related* examples as *agree*, *disagree*, or *discuss*.

3.3 Model Parameters and Training Procedure

For our CNN, we use 300-dimensional `Word2Vec` [11, 12, 13] word embeddings trained on `GoogleNews` dataset[3], and 128 feature maps with filter width {2, 3, 4}. We set maximum word lengths of 50 and 500 for claims and documents respectively; these values are greater than the length for most claims and documents in the target train data. For the BOW model, we keep the hyper-parameters and features the same as the baseline model [16]. Our models are trained using the `Adam` optimizer, and 20% of the training data is set aside as validation data. In the models with a domain adaptation (DA) component, equal amounts of both *source* and *target* data are randomly selected at each epoch during training. Finally, we fine-tune all the hyper-parameters of our models on validation data which contains equal amounts of *source* and *target* data.

3.4 Baselines

We compare our domain adaptation (DA) model to the following baselines: (a) *Gradient Boosting*, which is the Fake News Challenge baseline that trains a gradient boosting classifier using hand-crafted features reflecting polarity, refute, similarity and overlap between documents and claims; (b) *TALOS* [1], which was ranked first at FNC. It uses hand-crafted features as well as weighted-average between gradient-boosted decision trees (TALOS-Tree) and a deep convolutional neural network (TALOS-DNN); (c) *UCL* [16], which was ranked third at FNC. This model trains a softmax layer using n-gram features (e.g., TF and TF.IDF). We compare with this model as our BOW model is similar to this model and uses the same features.

3.5 Our Models

We present different variations of our models where each uses a subset of components/features shown in Figure 1. These variations help us to conduct ablation analyses on this information. The baseline and our models are trained to predict stance labels on target data; {*agree*, *disagree*, *discuss*, *unrelated*}. We further apply a two-level *hierarchy* prediction scheme in our models. That is, a model with a *hierarchical* scheme is first trained to predict two stance labels as {*unrelated* or *related*}, and then the examples predicted as *related* are only given to the model to predict the labels {*agree*, *disagree*, *discuss*}. For the first step of the *hierarchy* scheme, we use the BOW model which achieves a high F_1 performance of 97.7% for *unrelated* and 93.9% for *related* labels.

[2] The scorer is available at `www.fakenewschallenge.org`

[3] `https://code.google.com/p/word2vec`

Table 1: Results on the FNC test data. BOW, CNN and DA refer to our model when it uses bag-of-words features, convolutional features, and domain adaptation, respectively. When DA is present, square brackets indicate which features are passed to the domain adaptation component. The *hierarchy* in parentheses refers to our model with two-level prediction scheme as explained in section 3.5. We show the results of the models based on the smallest loss for validation set across 5 independent runs.

Model		**Train Data**	**Weighted-Acc.**	**Acc.**	**Macro-F$_1$**	**F$_1$** (*agree, disagree, discuss, unrelated*)
1.	Gradient Boosting	FNC	75.2	85.4	45.7	14.8 / 2.0 / 69.5 / 96.5
2.	TALOS *(#1st in FNC)*	FNC	82.0	89.1	57.8	53.8 / 3.6 / 76.0 / 97.9
3.	TALOS-DNN	FNC	60.8	66.5	41.8	27.6 / 9.3 / 47.4 / 82.7
4.	TALOS-Tree	FNC	83.1	89.5	56.8	53.4 / 0.2 / 76.3 / 98.4
5.	UCL *(#3rd in FNC)*	FNC	81.7	88.5	57.9	47.9 / 11.4 / 74.7 / 97.6
6.	BOW	FNC	81.1	88.6	56.0	49.2 / 2.5 / 74.8 / 97.6
7.	CNN	FNC	40.8	71.3	23.3	0.3 / 0.0 / 10.0 / 83.0
8.	BOW + CNN	FNC	74.9	86.8	52.2	41.4 / 0.0 / 72.1 / 95.2
9.	BOW (hierarchy)	FNC	80.7	88.5	57.3	49.5 / 7.5 / 74.3 / 97.7
10.	CNN (hierarchy)	FNC	79.9	87.9	56.0	54.9 / 0.2 / 71.1 / 97.7
11.	BOW + CNN (hierarchy)	FNC	80.3	88.2	56.5	56.0 / 0.0 / 72.1 / 97.7
12.	BOW	FNC, FEVER	78.5	86.4	56.3	48.8 / 9.8 / 69.4 / 97.1
13.	[BOW + DA]	FNC, FEVER	72.9	81.5	48.6	44.5 / 2.0 / 51.7 / 96.1
14.	BOW (hierarchy)	FNC, FEVER	78.8	87.3	57.2	51.7 / 10.2 / 69.1 / 97.7
15.	[BOW + DA] (hierarchy)	FNC, FEVER	78.5	87.1	56.4	51.6 / 8.3 / 68.0 / 97.7
16.	CNN	FNC, FEVER	41.3	64.3	27.3	17.4 / 2.0 / 11.0 / 78.8
17.	[CNN + DA]	FNC, FEVER	39.0	64.3	24.0	13.5 / 0.2 / 3.3 / 78.9
18.	CNN (hierarchy)	FNC, FEVER	79.0	87.4	56.6	51.9 / 7.5 / 69.3 / 97.7
19.	[CNN + DA] (hierarchy)	FNC, FEVER	79.1	87.7	57.9	51.2 / 11.4 / 71.3 / 97.7
20.	BOW + CNN	FNC, FEVER	71.7	84.5	51.5	44.6 / 5.6 / 60.2 / 95.6
21.	BOW + [CNN + DA]	FNC, FEVER	71.9	84.6	51.4	44.9 / 4.4 / 60.6 / 95.6
22.	BOW + CNN (hierarchy)	FNC, FEVER	79.6	87.8	56.6	53.1 / 5.1 / 70.6 / 97.7
23.	BOW + [CNN + DA] (hierarchy)	FNC, FEVER	**80.3**	**88.2**	**60.0**	**54.6 / 15.1 / 72.6 / 97.7**

3.6 Performance Analysis

Table 1 shows the results of different models for the target test data, i.e., FNC test data. Lines 1-5 show the results for baseline models explained in section 3.4. As the results show, they weakly perform for *agree* and *disagree* stances due to the small size of labeled data; only 1.7% and 7.3% of the FNC data has *disagree* and *agree* labels. Lines 6-23 show the results of different configurations of our model as explained in section 3.5. Lines 6-8 show the results for BOW, CNN and their combinations respectively; the results of these models with the *hierarchy* scheme are shown in lines 9-11. The results show that using the *hierarchy* scheme can help models to perform better, especially for CNN model where its result improves from 23.3% F$_1$ (line 7) to 56% (line 10). However, these models don't improve the baseline results, except for the combination of BOW and CNN models with the *hierarchy* scheme (see line 11). This model achieves the best result for *agree*, while it poorly performs on *disagree*. These results show that a promising solution is to use domain adaptation to deal with the limited data size as we show below.

While the above results (lines 1-11) are obtained when only the FNC train data is used, lines 12-23 show the results of different configuration of our model when we use additional training data from a different domain (i.e., FEVER) in addition to the FNC data. Lines 12-15 show the results for the BOW model with and without the *hierarchy* scheme and domain adaptation (DA) component. The results show that domain adaptation does not improve the results for BOW model. We repeat these experiments with CNN model (lines 16-19), and the CNN model with *hierarchy* scheme can significantly perform better when using the DA component. Given these results, we pass only convolutional features to the DA component in lines 20-23 which show the results for our model when using a combination of BOW and CNN model with and without the *hierarchy* scheme and domain adaptation. The model that combines BOW and CNN model with the *hierarchy* scheme and domain adaptation achieves the best F$_1$ performance, especially for *disagree* and *agree* classes—because the *source* data only contains the corresponding *disagree* and *agree* labels. In summary, the results show:

- The *hierarchy* scheme can help our models to perform better across all the metrics.
- Our best model is the combination of BOW, CNN+DA, and the hierarchy scheme. It outperforms the baselines on F$_1$, especially on the most important classes: *disagree*, *agree*.
- The *source* FEVER data can improve the performance of our model for target FNC data through adversarial domain adaptation, when uses CNN model with *hierarchy* scheme (see lines 18-19 in Table 1) or BOW+CNN model (see lines 20-23 in Table 1).

3.7 Training Loss Trend

Figure 2 shows the classification and domain adaptation losses across epochs on validation data for our best model: BOW + CNN + DA model with the *hierarchy* scheme. As shown in the figure, both losses are unstable during the early epochs of training. This is because of learning and domain adaptation rates, which couples a high learning rate with a ramp up in the domain adaption rate. Then, after around 10 epochs, the classification and domain adaptation losses get more stable and behave as expected; the classification loss slowly decreases as the label prediction component in the model attempts to minimize its loss, while the domain adaptation loss slowly increases as the model attempts to maximize the domain loss so that it cannot distinguish between its *source* and *target* examples as we explained in section 2.

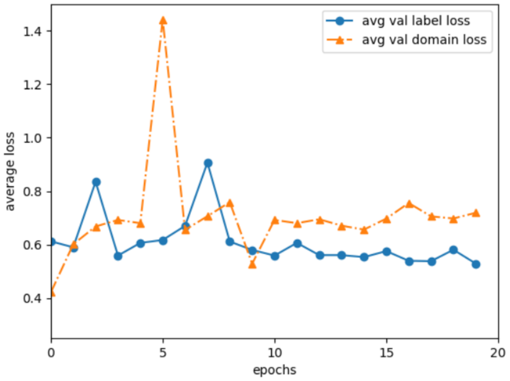

Figure 2: The classification (or label) and domain adaptation losses on validation data across training epochs for our best model; BOW + CNN + DA with *hierarchy* scheme.

4 Conclusion

We present a model that uses adversarial domain adaptation for the task of stance detection. Our experiments show the effectiveness of our model in transferring knowledge among stance datasets, from the FEVER dataset to the FNC dataset. Our model outperforms the state-of-the-art approaches for this task and obtains 60% F_1 on the FNC target data. For future work, we plan to apply our best model with adversarial domain adaptation to other combinations of datasets collected for stance detection. Furthermore, we plan to use the datasets collected for a similar task, e.g., the Stanford Natural Language Inference (SNLI) data [3] to investigate the utility of our model in transfer learning between inference and stance detection tasks.

Acknowledgments

This research was supported in part by the HBKU Qatar Computing Research Institute.

References

[1] Sean Baird, Doug Sibley, and Yuxi Pan. Talos targets disinformation with fake news challenge victory. In *Fake News Challenge*, 2017.

[2] Ramy Baly, Mitra Mohtarami, James Glass, Lluís Màrquez, Alessandro Moschitti, and Preslav Nakov. Integrating stance detection and fact checking in a unified corpus. In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics*, New Orleans, LA, USA, 2018.

[3] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2015.

[4] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32Nd International Conference on Machine Learning - Volume 37*, ICML'15, pages 1180–1189. JMLR.org, 2015.

[5] Andreas Hanselowski, Avinesh PVS, Benjamin Schiller, and Felix Caspelherr. Team Athene on the fake news challenge. https://medium.com/@andre134679/team-athene-on-the-fake-news-challenge-28a5cf5e017b, 2017.

[6] Georgi Karadzhov, Preslav Nakov, Lluís Màrquez, Alberto Barrón-Cedeño, and Ivan Koychev. Fully automated fact checking using external sources. In *Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP)*, pages 344–353, 2017.

[7] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751, 2014.

[8] Todor Mihaylov, Georgi Georgiev, and Preslav Nakov. Finding opinion manipulation trolls in news community forums. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 310–314. Association for Computational Linguistics, 2015.

[9] Todor Mihaylov and Preslav Nakov. Hunting for troll comments in news community forums. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 399–405, Berlin, Germany, 2016.

[10] Tsvetomila Mihaylova, Preslav Nakov, Lluis Marquez, Alberto Barron-Cedeno, Mitra Mohtarami, Georgi Karadzhov, and James Glass. Fact checking in community forums. In *Proceedings of AAAI*, New Orleans, LA, USA, 2018.

[11] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *CoRR*, abs/1301.3781, 2013.

[12] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. *CoRR*, abs/1310.4546, 2013.

[13] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic Regularities in Continuous Space Word Representations. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 746–751, 2013.

[14] Mitra Mohtarami, Ramy Baly, James Glass, Preslav Nakov, Lluís Màrquez, and Alessandro Moschitti. Automatic stance detection using end-to-end memory networks. In *Proceedings of the 16th Annualw Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL-HLT '18, New Orleans, LA, USA, 2018.

[15] Mitra Mohtarami, Yonatan Belinkov, Wei-Ning Hsu, Yu Zhang, Tao Lei, Kfir Bar, Scott Cyphers, and James Glass. Sls at semeval-2016 task 3: Neural-based approaches for ranking in community question answering. In *Proceedings of NAACL-HLT Workshop on Semantic Evaluation*, pages 753–760, San Diego, California, June 2016. Association for Computational Linguistics.

[16] Benjamin Riedel, Isabelle Augenstein, Georgios P. Spithourakis, and Sebastian Riedel. A simple but tough-to-beat baseline for the fake news challenge stance detection task. *CoRR*, abs/1707.03264, 2017.

[17] Aliaksei Severyn and Alessandro Moschitti. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 373–382. ACM, 2015.

[18] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Fever: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819. Association for Computational Linguistics, 2018.

[19] Andreas Vlachos and Sebastian Riedel. Fact checking: Task definition and dataset construction. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pages 18–22, Baltimore, MD, USA, 2014.

[20] Soroush Vosoughi, Deb Roy, and Sinan Aral. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018.

[21] Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. Deep learning for answer sentence selection. *CoRR*, abs/1412.1632, 2014.