From Fake to Hyperpartisan News Detection Using Domain Adaptation

Răzvan-Alexandru Smădu[1], **Sebastian-Vasile Echim**[1], **Dumitru-Clementin Cercel**[1], **Iuliana Marin**[2], **Florin Pop**[1,3]
[1]University Politehnica of Bucharest, Faculty of Automatic Control and Computers
[2]University Politehnica of Bucharest, Faculty of Engineering in Foreign Languages
[3]National Institute for Research & Development in Informatics - ICI Bucharest, Romania
razvan.smadu@stud.acs.upb.ro, sebastian.echim@stud.aero.upb.ro
{dumitru.cercel,iuliana.marin,florin.pop}@upb.ro

Abstract

Unsupervised Domain Adaptation (UDA) is a popular technique that aims to reduce the domain shift between two data distributions. It was successfully applied in computer vision and natural language processing. In the current work, we explore the effects of various unsupervised domain adaptation techniques between two text classification tasks: fake and hyperpartisan news detection. We investigate the knowledge transfer from fake to hyperpartisan news detection without involving target labels during training. Thus, we evaluate UDA, cluster alignment with a teacher, and cross-domain contrastive learning. Extensive experiments show that these techniques improve performance, while including data augmentation further enhances the results. In addition, we combine clustering and topic modeling algorithms with UDA, resulting in improved performances compared to the initial UDA setup.

1 Introduction

Fake news detection is a challenging task in which the goal is to detect whether the news content does not disseminate false information which may harm society. Recently, this problem has broad attention to the research community, especially with the rising interaction with social media platforms, which have become one of the primary sources of information for many individuals (Shu et al., 2020). Detecting fake news is challenging for many of us, since some news can be written very convincingly, thus spreading misleading information without control (Ahmed et al., 2017). Therefore, new datasets (such as BuzzFeed-Webis Fake News (BuzzFeed) (Potthast et al., 2018) and ISOT (Ahmed et al., 2017)) and novel detection techniques (Koloski et al., 2022; Mosallanezhad et al., 2022) have emerged in recent years.

Especially since the 2016 United States presidential election, a related task, namely hyperpartisan news detection, identifies whether the information spread by the news is in a political extreme (Rae, 2021). Hyperpartisan articles aim to expose information related to only one perspective, ignoring and, in some cases, even attacking the perspectives from other opposing sides (Kiesel et al., 2019). The consequences of this type of news range from misinformation in the media to an increase in the number of supporters of extreme ideologies (Huang and Lee, 2019).

Some works (Potthast et al., 2018; Ross et al., 2021) linked fake news with hyperpartisan news, since their goal is to spread as much as possible and influence people. This phenomenon is related to clickbait (Potthast et al., 2016), as the authors use different techniques to make the content more accessible and viral on the media (Kiesel et al., 2019).

Recently, many architectures based on Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) have been developed and fine-tuned on various natural language processing (NLP) tasks. The current work aims to evaluate unsupervised deep learning techniques on the fake news detection task and adapt them to the hyperpartisan news detection task. Specifically, we employ the Robustly optimized BERT pretraining approach (RoBERTa) (Liu et al., 2019) and evaluate it in three domain adaptation scenarios: unsupervised domain adaptation (UDA) (Ganin and Lempitsky, 2015), cluster alignment with a teacher (CAT) (Deng et al., 2019), and cross-domain contrastive learning (CDCL) (Chen et al., 2020). In addition, we analyze topic modeling and clustering algorithms to generate domain labels and perform UDA to learn about topic-aware features which are specific to fake and hyperpartisan news detection. More precisely, we evaluate various clustering algorithms for generating domain labels, namely K-Means (Lloyd, 1982), K-Medoids (Kaufmann,

Proceedings of Recent Advances in Natural Language Processing, pages 1095–1109
Varna, Sep 4–6, 2023
https://doi.org/10.26615/978-954-452-092-2_117

1987), Gaussian Mixture (Fraley and Raftery, 2002), and HDBSCAN (Campello et al., 2013). Additionally, we explore four topic modeling algorithms: Latent Dirichlet Allocation (LDA) (Blei et al., 2003), Non-negative Matrix Factorization (NMF) (Lee and Seung, 1999), Latent Semantic Analysis (LSA) (Deerwester et al., 1990), and probabilistic LSA (pLSA) (Hofmann, 1999).

Therefore, the main contributions of this work are as follows:

- We evaluate the RoBERTa model on a domain adaptation from fake to hyperpartisan news detection by comparing three techniques, as well as several fine-tuning strategies.
- To our knowledge, we are the first to show that cross-domain contrastive learning proposed by Wang et al. (2022), initially employed on computer vision, which performs better than other unsupervised learning techniques on an NLP task.
- We propose the cluster and topic-based UDA approaches, which obtain better results when compared with the original formulation for UDA.
- We perform extensive experiments to assess the effectiveness of each employed method under various hyperparameter configurations and data augmentation techniques based on the term frequency-inverse document frequency (TF-IDF) scores (Salton et al., 1975) and the Generative Pre-trained Transformer 2 (GPT-2) model (Radford et al., 2019).

2 Related Work

2.1 Fake News Detection

Machine learning techniques for detecting fake news include various feature-based methods, ranging from text to visual features (Zhang and Ghorbani, 2020). For example, linguistic features (Choudhary and Arora, 2021; Pérez-Rosas et al., 2018) capture aspects related to conveyed information, document organization, and vocabulary used in news. In contrast, style-based features (Potthast et al., 2018; Zhou and Zafarani, 2020) are related to the writing style, such as redaction objectivity and deception (Shu et al., 2017). In recent years, Transformer-based models (Vaswani et al., 2017) emerged in the fake news detection literature (Jwa et al., 2019; Zhang et al., 2020; Kaliyar et al., 2021; Szczepański et al., 2021). Other techniques for detecting fake news use social aspects, such as the profiles of the users who spread the news on social media platforms (Shu et al., 2017; Onose et al., 2019; Zhou and Zafarani, 2020; Sahoo and Gupta, 2021). Techniques successfully employed for these scenarios rely on custom embeddings and linear classifiers (Shu et al., 2019), classic supervised machine learning techniques (Reis et al., 2019), and deep learning networks, such as recurrent (Wu and Liu, 2018) and graph neural networks (Monti et al., 2019; Hamid et al., 2020; Paraschiv et al., 2021).

2.2 Hyperpartisan News Detection

Task 4 of SemEval-2019 (Kiesel et al., 2019) introduced hyperpartisan detection from news articles as a binary classification task. The organizers created two balanced datasets by crawling data from various online publishers. Participants were asked to detect whether the news articles were hyperpartisan or mainstream. The winning team (Jiang et al., 2019) of the shared task proposed an architecture based on multiple pre-trained ELMo embeddings (Peters et al., 2019) averaged in the embedding space, followed by convolutional layers (Kim, 2014) and batch normalization (Ioffe and Szegedy, 2015). They achieved 84.04% accuracy on the training set and 82.16% accuracy on the test set, suggesting the challenging setting. Other works for the SemEval-2019 Task 4 were based on lexical and semantic handcrafted features via Universal Sentence Encoder (Cer et al., 2018) or BERT, and a linear classifier (Srivastava et al., 2019; Hanawa et al., 2019). Furthermore, Potthast et al. (2018) showed that hyperpartisan news detection could be analyzed using fake news approaches. They argued that the writing style for hyperpartisan news is similar to fake news, despite their political orientation.

2.3 Unsupervised Domain Adaptation

The core objective of unsupervised domain adaptation is to enforce a feature representation invariant to the domain of the examples with the same labels. One of the most effective techniques is the work of Ganin and Lempitsky (2015), which treated the problem as a minimax optimization. Wang et al. (2018) utilized domain adaptation techniques via adversarial training for fake news detection by employing an event discriminator to learn event-invariant features in a multi-modal setting. Deng et al. (2019) relied on the similarity in the feature space by enforcing a clustered structure among similar features. In this case, the training procedure optimizes clustering loss alongside the domain adaptation loss. For the target dataset, a teacher model consisting of an ensemble of students generates pseudo-labels (i.e., estimates of the true labels). Also, contrastive learning (Chen et al., 2020) was used to achieve unsupervised domain adaptation. It aims to have closer representations of the examples from the same class, while representations from different classes should stay far apart. In addition, Wang et al. (2022) proposed the cross-domain contrastive loss to minimize the l_2-norm distance between features from the same category, and employed K-Means to compute pseudo-labels.

3 Method

3.1 Base Model

In our current work, we utilize the pre-trained RoBERTa language model, which shares the same architectural design as BERT, the only difference being the pre-training objectives. The RoBERTa architecture stacks multiple Transformer encoders, each based on the multi-head self-attention mechanism (Vaswani et al., 2017). On top of the RoBERTa model, we add a label predictor containing fully connected layers. RoBERTA uses the Byte-Pair Encoding (BPE) tokenizer (Sennrich et al., 2015). In what follows, we present the settings in which RoBERTa is employed in our work (see Figure 1).

3.2 Unsupervised Domain Adaptation

Given two datasets $D_s = \{(x_s^i, y_s^i)\}_{i=1}^{N_s}$ and $D_t = \{x_t^i\}_{i=1}^{N_t}$ from different domains, the UDA setting reduces the shift between them (Ganin and Lempitsky, 2015; Ganin et al., 2016). This approach comprises a feature encoder G_f, a label predictor G_y, and a domain discriminator G_d. The feature encoder maps the input space into a latent space. Then, the label predictor computes the labels of the underlying examples. Simultaneously, the domain classifier uses the latent space to predict the domain of the features (i.e., the source or target domain).

To obtain domain-invariant features, the optimization is two-fold. First, we minimize the prediction loss concerning G_f's parameters θ_f and G_y's parameters θ_y. Second, we maximize the domain classification loss until G_d cannot distinguish the domains of the features. Formally, the loss function L (see Eq. 1) depends on the prediction loss L_y between G_y's outputs and source labels, and the domain adaptation loss L_d between G_d's outputs and domains d^i (i.e., hyperpartisan and fake news). The trade-off between L_y and L_d is controlled by λ. Note that we omitted the model's parameters for clarity.

$$\begin{aligned} L = &\sum_{i=1}^{N_s} L_y(G_y(G_f(x_s^i)), y_s^i) \\ &- \lambda \sum_{i=1}^{N} L_d(G_d(G_f(x^i)), d^i) \end{aligned} \tag{1}$$

The optimization problem associated with this formulation is described below:

$$\hat{\theta}_f, \hat{\theta}_y = \arg\min_{\theta_f, \theta_y} L(\theta_f, \theta_y, \hat{\theta}_d) \tag{2}$$

$$\hat{\theta}_d = \arg\max_{\theta_d} L(\hat{\theta}_f, \hat{\theta}_y, \theta_d) \tag{3}$$

where the parameters with hat are fixed during the optimization step. This problem can be solved with an implementation trick, namely gradient reversal layer (GRL) (Ganin and Lempitsky, 2015), which acts as the identity function during feed-forward and negates the gradients during back-propagation. The GRL layer is inserted between the feature encoder and the domain discriminator.

In our setting, we use the RoBERTa's encoders for feature extraction and fully connected layers for both the label predictor and domain discriminator.

3.3 Cluster Alignment with a Teacher

As an extension to UDA, Deng et al. (2019) exploited the class-conditional structure of the feature space by cluster alignment in the teacher-student paradigm. A teacher model trained on the labeled source examples estimates pseudo-labels for the unlabeled target dataset. To reduce the error amplification caused by label estimation, the teacher model is built as an ensemble of previous student classifiers. In addition, a student classifier minimizes the prediction loss L_y on the source examples in the supervised setting. The optimization involves minimizing both the prediction loss L_y and the sum of clustering losses L_c (i.e., for both the source and the target domains) and the cluster-base alignment loss L_a:

$$L = L_y + \alpha(L_c + L_a) \tag{4}$$

where the hyperparameter α controls the trade-off between the supervised and semi-supervised losses.

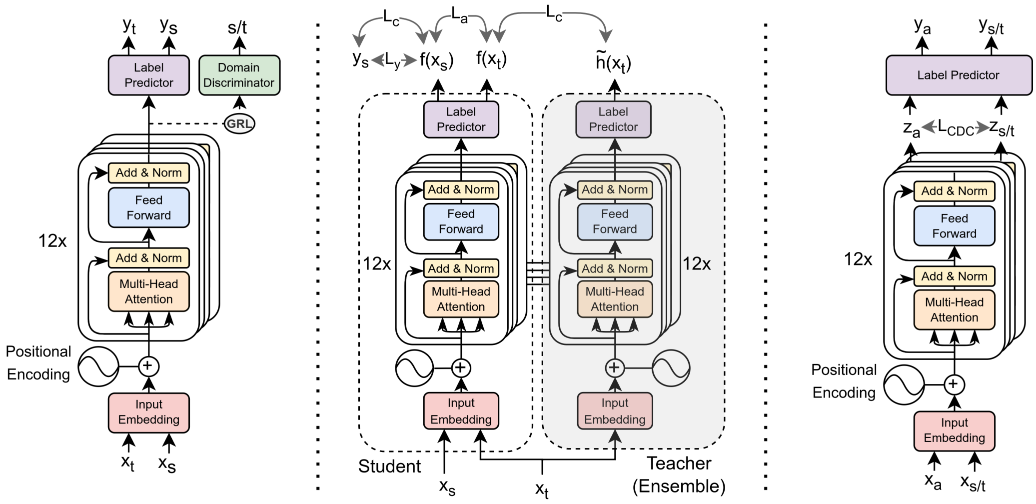


Figure 1: (Left) The RoBERTa model in the UDA setting includes a label predictor and a domain discriminator. (Center) In the CAT method, the student and teacher use the RoBERTa model. (Right) In the CDCL setting, the contrastive loss is applied between the RoBERTa features of an anchor and the source (s) / target (t) example.

Considering the labeled samples $X_s = \{x_s^i, y_s^i\}_{i=1}^{N_s}$, the unlabeled samples $X_t = \{x_t^i\}_{i=1}^{N_t}$, the feature extractor $f(\cdot)$, and the distance metric d between features, the total clustering loss is:

$$L_c(X_s, X_t) = L_c(X_s) + L_c(X_t) \tag{5}$$

where L_c is as follows for each X_*:

$$L_c(X_*) = \frac{1}{|X_*|^2} \sum_{i=1}^{|X_*|} \sum_{j=1}^{|X_*|} [\delta_{ij} d(f(x^i), f(x^j)) + (1 - \delta_{ij}) \max(0, m - d(f(x^i), f(x^j)))] \tag{6}$$

The intuition is to enforce class-conditional structure at the feature representation by grouping the classes into clusters, i.e., by minimizing the distance between features x^i and x^j that have the same label when the indicator function $\delta_{ij} = 1$, whereas pushing different clusters away from at least a margin m by maximizing the feature distance when $\delta_{ij} = 0$. The classifier trained on the source features may not be able to differentiate between the same class from different domains, and therefore, an alignment loss L_a is imposed between the domains as follows:

$$L_a(X_s, X_t) = \frac{1}{K} \sum_{k=1}^{K} ||\lambda_{s,k} - \lambda_{t,k}||_2^2 \tag{7}$$

In this case, given the number K of classes to be predicted, and the samples $X_{*,k}$ from either source or target whose labels are equal to k, the cluster centroids $\lambda_{*,k}$ are computed using:

$$\lambda_{*,k} = \frac{1}{|X_{*,k}|} \sum_{x_*^i \in X_{*,k}} f(x_*^i) \tag{8}$$

The loss L_a tries to match the source and target statistics by aligning the clusters for each class k in the feature space. Additionally, the performance can be further improved by aligning the marginal distributions, i.e., adding a confidence threshold that ignores the data points likely to be included in the wrong class.

3.4 Cross-Domain Contrastive Learning

Self-supervised contrastive learning (Chen et al., 2020) aims to learn representations such that, given a pair of examples, closely related examples should behave similarly, while dissimilar examples should stay far apart from each other. This can be achieved by employing various techniques such as data augmentation and custom losses (e.g., NT-Xent (Chen et al., 2020), InfoNCE (Oord et al., 2018)). Since there is no clear way to construct positive and negative pairs in an unsupervised domain adaptation framework, Wang et al. (2022) argued that samples from the same category should be similar. In contrast, samples from different categories should have other feature representations, regardless of the domain from which they come. Based on this hypothesis, they proposed the cross-domain contrastive (CDC) loss to reduce the domain shift between source and target labels. We assume z_t^a and

z_s^p are the l_2-normalized features for the anchor sample from the target domain x_t^a and the positive sample from the source domain x_s^p, respectively. In this case, the loss function is described by:

$$L_{CDC}^{t,a} = -\frac{1}{|P_s(\hat{y}_t^a)|} \sum_{p \in P_s(\hat{y}_t^a)} log \frac{\exp(z_t^a \cdot z_s^p / \tau)}{\sum_{j \in I_s} \exp(z_t^a \cdot z_s^j / \tau)} \quad (9)$$

where $P_s(\hat{y}_t^a)$ denotes the set of positive samples from the source domain having the same label as the anchor point, and I_s is the set of all source samples from the mini-batch. Similar to Eq. 9, we compute $L_{CDC}^{s,a}$, for which we consider the positive samples from the target domain instead. The CDC loss with alignment at the feature level is[1]:

$$L_{CDC} = \frac{1}{N_s} \sum_{a=1}^{N_s} L_{CDC}^{s,a} + \frac{1}{N_t} \sum_{a=1}^{N_t} L_{CDC}^{t,a} \quad (10)$$

The objective function is given by the sum of the prediction loss L_y and the loss L_{CDC} scaled by γ:

$$L = L_y + \gamma L_{CDC} \quad (11)$$

We generate pseudo-labels using the K-Means algorithm since we require them when creating positive pairs. We initialize K-Means with the centroids of the source domain and predict on the target domain. The pseudo-labels are chosen to minimize the similarity distance between the feature representation and the centroid. K-Means is performed at the beginning of each epoch.

3.5 Cluster and Topic-Based Unsupervised Domain Adaptation

We propose an addition to the UDA approach, considering the supervised setting (i.e., we have access to the labeled source dataset). First, we represent the input text using TF-IDF or a pre-trained RoBERTa model. We employ a clustering/topic modeling algorithm in this feature space to identify k clusters or topics, which will be assigned as domain labels. For clustering, we employ four algorithms, namely K-Means, K-Medoids, Gaussian Mixture, and HDBSCAN. Also, we use four topic modeling algorithms, namely LDA, NMF, LSA, and pLSA. The motivation is to compact the latent representation, given estimates of latent domains under a topic model (i.e., a dataset split). During training, it is minimized the loss given by Eq. 1 while using the proposed domain labels. For the target examples, we do not include labels during training. We choose the number of clusters using the elbow method[2]. After training on each pair of domain labels, the best-performing model is selected for the inference stage.

4 Experimental Setup

4.1 Datasets

We perform experiments on three datasets related to fake (i.e., ISOT and BuzzFeed) and hyperpartisan (i.e., BuzzFeed and Hyperpartisan (Kiesel et al., 2019)) news detection.

The ISOT fake news dataset contains news articles collected from reuters.com, and other websites, which were validated by Politifact[3]. The dataset comprises 44,898 articles, of which 21,417 contain truthful information, and 23,481 are fake news. All collected articles are related to politics and have at least 200 characters.

The BuzzFeed dataset contains 1,627 articles in three categories: mainstream, left-wing, and right-wing. The mainstream and hyperpartisan data are evenly distributed, and the length of the articles ranges between 400 and 800 words. This dataset is annotated for both fake and hyperpartisan news detection.

The Hyperpartisan dataset which contains hyperpartisan news was released under the SemEval-2019 Task 4 shared task (Kiesel et al., 2019). The dataset was crawled from news publishers listed by BuzzFeed[4] and Media Bias Fact Check[5]. From these sources, 754,000 news articles were extracted and semi-automated labeled using distant supervision (Mintz et al., 2009) at the publisher level, provided in the HTML format. It was split into 600,000 articles for training, 150,000 articles for validation, and 4,000 articles for testing. Half of the dataset consists of non-hyperpartisan articles, and the other half is split equally among left-wing and right-wing articles. Since the authors also released a smaller version of the dataset (645 examples for training and 628 examples for testing), in what follows, we will refer to the larger

[1] Note that we included the normalization terms compared to the original formulation.

[2] https://www.scikit-yb.org/en/latest/api/cluster/elbow.html

[3] An organization that checks the veracity of the news.

[4] https://github.com/BuzzFeedNews/2017-08-partisan-sites-and-facebook-pages

[5] https://mediabiasfactcheck.com

dataset as Hyperpartisan-L and the smaller dataset as Hyperpartisan-S.

4.2 Data Preprocessing

We perform data cleaning on all three corpora, ignoring non-ASCII characters and removing HTML-specific symbols and constructions that do not provide any information about the actual content, such as multiple chains of dots in a line. BPE was utilized for tokenization, setting to output a maximum of 128 tokens per text sample.

Since the ISOT and BuzzFeed datasets are not provided with separate splits for validation and testing, we use the following split: 70% for training, 10% for validation, and 20% for testing. In addition, due to limited computational resources and the large size of the Hyperpartisan dataset, we select a random 5% of the data from the training set (i.e., 30,000 examples) and 5% of the data for the validation set (i.e., 7,500 examples). Also, we use the entire Hyperpartisan test set since it contains only 4,000 examples.

4.3 Hyperparameters

We utilize the pre-trained RoBERTa base version (123M parameters), which consists of a stack of 12 Transformer blocks. For all experiments, the Adam optimizer (Kingma and Ba, 2015) with a linear scheduler is used with a warm-up (it is set with 5% of the gradient steps) for the learning rate. The learning rate varies among experiments, between $1e-4$ and $1e-5$. We employ a dropout set between 0.1 and 0.5. We also set the optimizer's weight decay parameter to $1e-3$, and clip the gradients between -1 and 1 to increase training stability and reduce overfitting.

5 Results

There were conducted multiple experiments to evaluate the impact of using various fine-tuned models for RoBERTa. We also investigate the effects of fine-tuning the RoBERTa model on the downstream task. Then, we analyze the impact of using a data augmentation technique (Xie et al., 2020) based on the TF-IDF scores. In Appendix A.1, we present the results of the GPT-2 data augmentation. Finally, we use clustering and topic modeling algorithms to extract clusters and topics from the training set and perform domain adaptation. We present the results in terms of accuracy (Acc) and F1-score (F1).

Dataset	Acc(%)	F1(%)
BuzzFeed	96.9	96.7
ISOT	99.8	99.7
Hyperpartisan-S	83.7	83.0
Hyperpartisan-L	62.1	69.0

Table 1: Results obtained after fine-tuning and evaluating RoBERTa on each dataset.

Model	Acc(%)	F1(%)
RoBERTa	62.1	69.0
RoBERTa frozen	53.7	65.4
RoBERTa fine-tuned first on BuzzFeed	62.3	68.0
RoBERTa fine-tuned first on ISOT	**63.0**	**70.0**

Table 2: Results for different fine-tuning strategies on the Hyperpartisan-L dataset.

5.1 Baselines

We start with the most straightforward approach for training a neural network. That is, we take a pre-trained model on similar tasks and transfer some of the acquired knowledge to the downstream task via fine-tuning. The baseline model consists of the RoBERTa model followed by a stack of fully connected layers. We employ two fully connected layers, with 256 hidden units and two output neurons. The models are trained for 3 epochs, with a learning rate of $1e-4$ and batch size between 32 and 64.

First, we evaluate the model on all four datasets for baseline results. Table 1 presents the final results obtained during experiments. We observe that ISOT achieves the highest scores, followed by BuzzFeed and Hyperpartisan-S. We note that humans annotated these datasets, whereas the Hyperpartisan-L dataset was annotated with a semi-supervised approach.

By comparing three fine-tuning methods (see Table 2), we observe that freezing the model's encoders yields poor performance. This increases the number of false positives and decreases the number of true negatives because of the domain shift between the datasets and training with fewer parameters. On the other hand, fine-tuning improves the results since the model's parameters are adapted to the new domain.

5.2 Results for UDA

We consider the encoders from the RoBERTa model as feature generators. We also use a stack of fully connected layers, with 256 hidden neurons and two outputs for both the label predictor and the

λ	Source	Target	Source		Target	
			Acc(%)	F1(%)	Acc(%)	F1(%)
0.1	Hyperpartisan-L	BuzzFeed	**61.5**	67.7	**85.4**	**86.4**
1	Hyperpartisan-L	BuzzFeed	58.1	**68.4**	60.8	38.2
5	Hyperpartisan-L	BuzzFeed	50.0	2.5	54.0	3.8
0.1	BuzzFeed	Hyperpartisan-L	95.3	94.9	**64.3**	62.7
1	BuzzFeed	Hyperpartisan-L	**96.5**	**96.6**	50.0	**66.5**
5	BuzzFeed	Hyperpartisan-L	51.5	7.1	50.8	7.7
0.1	BuzzFeed	Hyperpartisan-L	94.4	94.5	56.7	64.1

Table 3: Unsupervised domain adaptation between Hyperpartisan-L and BuzzFeed datasets.

GRL pos.	Source		Target	
	Acc(%)	F1(%)	Acc(%)	F1(%)
4	**95.9**	**95.2**	62.1	61.7
6	95.0	94.4	62.1	**67.1**
10	91.3	89.1	60.9	64.1
12	95.3	94.9	**64.3**	62.7

Table 4: Various linking positions of the GRL layer to the encoders of RoBERTa, on BuzzFeed (source) to Hyperpartisan-L (target) adaptation.

domain discriminator. The domain discriminator is linked to the output of the RoBERTa encoder via a gradient reversal layer. We tested three values for $\lambda \in \{0.1, 1, 5\}$.

Furthermore, we perform larger-to-smaller and smaller-to-larger dataset adaptations between Hyperpartisan-L and BuzzFeed. The model is trained for 3 epochs (i.e., the steps required to pass through all examples from the larger dataset). The batch size is set to 64, from which half are labeled and the other half are unlabeled examples. The results are shown in Table 3. We observe that if λ is set too large, the model does not learn the data distribution but predicts only one class. Conversely, UDA performs better when $\lambda = 0.1$, achieving higher accuracy on the Hyperpartisan-L target dataset. This adaptation may have helped because of the inherent similarities between domains and improved performance on out-of-distribution points.

Moreover, we employ different ways of linking the GRL layer with the RoBERTa encoders. Since the RoBERTa-base model uses 12 encoders, we utilized the 4th, 6th, and 10th, besides the previous experiments. While the encoder returns a feature representation for each element in the sequence, we take the representation of the `[CLS]` token. Table 4 shows the results. The 12th layer performs best, while similar performances are achieved using the 4th or 6th layer. The results are supported by the fact that more layers for the encoder mean more representational power for the feature encoder that needs to be adapted among domains.

λ	α	Source		Target	
		Acc(%)	F1(%)	Acc(%)	F1(%)
1	1	92.5	91.2	51.3	**66.4**
1	0.1	94.7	93.8	57.9	62.6
0.1	0.1	95.9	95.7	**59.9**	61.5
0.1	0	**96.5**	**96.4**	58.7	64.3
0	0.1	95.6	95.4	59.8	64.1
0	0	93.7	92.7	58.9	62.5

Table 5: Results for the CAT framework on BuzzFeed (source) to Hyperpartisan-L (target) adaptation.

5.3 Results for CAT

In addition to the previous experimental setup, we set the parameter $\alpha \in \{0.1, 1\}$ for the clustering loss in the CAT configuration. We also consider a lower learning rate (i.e., $1e - 5$) to improve convergence. We consider an epoch is a complete pass through the smaller dataset to update the pseudo-labels for the entire target domain using the teacher model. As such, we trained the models for 10-30 epochs. We set the margin $m = 2$, the ensemble size to 3, and the ensemble accumulation to 0.8.

We performed domain adaptation from BuzzFeed to Hyperpartisan-L. The results are shown in Table 5. The model obtains over 90% accuracy on the source domain and is bounded by 66.4% on the target domain. This approach generally achieves a smaller accuracy than previous techniques, the best score being when $\lambda = \alpha = 0.1$. Also, we can observe that the difference between λ and α affects the performances. Analyzing the model predictions, we notice that using smaller values for λ and α yields a high number of false positives, while larger values increase the number of false negatives. Using $\lambda = 1$ and $\alpha = 0.1$ resulted in a biased model towards mainstream examples.

5.4 Results for CDCL

For the CDCL method, the experimental setup is similar to the one used for the CAT. We varied the temperature $\tau \in \{0.1, 0.5, 1\}$ and the coefficient $\gamma \in \{0, 0.1, 1, 5\}$. Table 6 provides the results of our analysis. We observe that both τ and γ affect the performance. The best results were attained when $\tau = 1$, and $\gamma = 5$, achieving 63.9% accuracy on the target domain, while $\tau = 0.5$ generates the best values on the source dataset. It proves that L_{CDC} performs some regularization on the source domain. We noticed that the models often produce a high false positive rate, affecting the recall more than the precision. In addition, training for more epochs, the model starts overfitting on

τ	γ	Source		Target	
		Acc(%)	F1(%)	Acc(%)	F1(%)
0.1	0	95.6	95.2	59.9	62.2
0.1	0.1	91.3	90.1	63.3	64.9
0.1	1	96.2	96.0	61.9	68.8
0.1	5	96.2	96.0	62.6	67.9
0.5	0	95.0	95.2	60.4	64.3
0.5	0.1	95.3	95.7	57.1	67.8
0.5	1	89.4	89.6	60.8	63.9
0.5	5	**96.5**	**96.4**	63.4	66.5
1	0	95.9	95.8	63.3	65.2
1	0.1	95.9	95.8	61.6	68.6
1	1	92.2	92.6	61.9	67.3
1	5	95.6	95.4	**63.9**	**69.2**

Table 6: Results for the CDCL framework on BuzzFeed (source) to Hyperpartisan-L (target) adaptation.

p	Source		Target	
	Acc(%)	F1(%)	Acc(%)	F1(%)
UDA				
0	94.0	93.4	59.1	**64.5**
0.5	95.8	95.5	**63.2**	62.7
0.1/0.2	94.7	94.5	57.3	61.5
0.1/0.2/0.3	**98.4**	**98.3**	61.3	46.9
CAT				
0	95.9	95.7	59.9	61.5
0.5	93.0	92.7	60.5	**65.2**
0.1/0.2	**98.8**	**98.8**	**62.7**	64.0
0.1/0.2/0.3	98.2	98.1	60.7	64.7
CDCL				
0	94.0	93.4	60.8	69.4
0.5	95.1	94.8	63.2	69.0
0.1/0.2	97.3	97.3	63.6	68.9
0.1/0.2/0.3	**98.8**	**98.8**	**64.4**	**69.4**

Table 7: Results for the TF-IDF-based data augmentation. The source is BuzzFeed and the target is Hyperpartisan-L.

both source and target domains while degrading the performance of the validation set.

5.5 Results for Text Augmentation Based on TF-IDF

We explore a data augmentation technique based on TF-IDF as proposed by Oord et al. (2018) for consistency training. Thus, we compute the TF-IDF score for every token from the corpus and associate it with the probability of it being changed. The words with the higher probability are replaced with non-keywords from the vocabulary to avoid changing the meaning of the text. The TF-IDF-based word replacement depends on a hyperparameter p that controls the level of augmentation enabled on the dataset. We vary p for our experiments to augment the BuzzFeed dataset with multiple augmentation levels. Table 7 shows the results for all training configurations, where two or three values per augmentation type indicate that we applied each value of p and concatenated the augmented examples over the original dataset. Also, zero suggests that only the unaltered dataset was used. Using more augmentations (e.g., $p \in \{0.1, 0.2, 0.3\}$) on the CDCL and CAT frameworks yields better overall results, while on UDA, using a much stronger augmentation (i.e., $p = 0.5$) leads to better results.

One problem with this data augmentation technique is that it may alter the text in a way that is not coherent anymore, specifically when many tokens are changed. The most frequent words may not always have the same meaning, so their contextualized representation is affected. Since the context defines the meaning of a word in language models, this augmentation changes the representation, especially on unlabelled data. Table 7 illustrates the issue on the target dataset. However, on the source dataset, the performance is not affected but generally improved.

5.6 Results for Cluster- and Topic-Based UDA

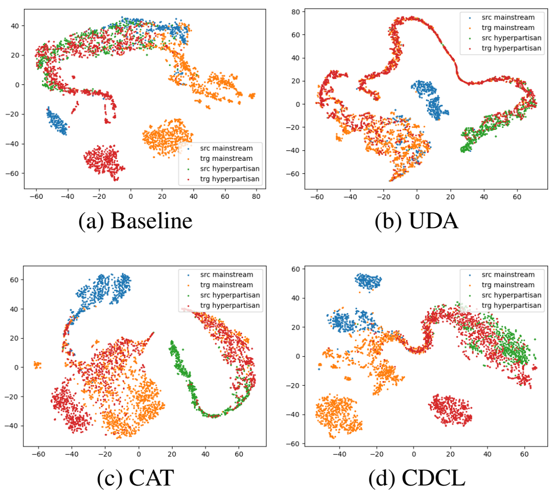
In the topic-based UDA approach, we follow the same experimental setup as in classical UDA. For training, the only difference is that we train all models for 10 epochs. We explore both, the clustering on RoBERTa features (i.e., K-Means with Euclidean or cosine distance, K-Medoids, Gaussian Mixture, and HDBSCAN) and the topic modeling algorithms on TF-IDF features (i.e., LDA, NMF, LSA, and pLSA) to split the representation. We evaluate the experiments on the Hyperpartisan-L test set and present the results in Table 8. Using clustering algorithms for domain labels provides the best overall results compared to Table 3. The best-performing models outperform the UDA approach by over 3% in accuracy and are obtained when we adapted from a larger to a smaller split. It is noteworthy that for the HDBSCAN, the cluster 2 contains very few annotated examples (i.e., 332) compared with the other two (i.e., 17,092 and 12,576), resulting in adaptation failure. When using the topic modeling, we see a degradation in performance, especially in the case of NMF. Compared with the RoBERTa baseline (see Table 2), the model achieves similar F1-scores.

5.7 Feature Visualization

We use t-SNE (Van der Maaten and Hinton, 2008) to visualize the feature representation learned by the best models we obtained for each category. In Figure 2, we present the plots for the baseline, the

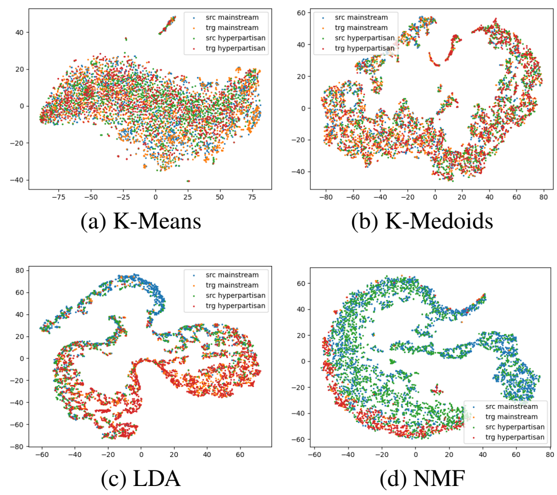
Method	0 → 1		1 → 0		2 → 0		0 → 2		1 → 2		2 → 1	
	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)
K-Means-euclidean	67.2	68.2	66.1	68.6	64.1	65.6	**67.9**	69.3	61.9	68.0	65.4	69.2
K-Means-cosine	64.2	69.0	63.5	70.0	66.0	63.6	66.3	67.8	64.1	68.5	62.4	67.3
K-Medoids	66.0	64.2	62.7	57.8	66.3	68.0	64.2	57.5	61.7	52.1	63.5	60.9
Gaussian Mixture	67.1	**70.6**	59.5	67.7	57.9	64.0	64.9	69.6	59.7	68.0	65.3	64.2
HDBSCAN	65.1	68.9	62.5	63.4	50	0.0	60.0	55.6	62.2	66.0	50.0	0.0
LDA	61.8	52.2	59.0	43.5	66.1	61.9	62.6	66.2	49.4	61.9	59.8	46.2
NMF	63.3	53.3	59.9	55.7	56.0	58.1	54.9	36.3	59.8	57.0	60.5	45.4
LSA	62.1	70.3	50.0	66.4	51.5	8.6	51.6	65.6	53.1	64.6	61.4	70.0
pLSA	61.6	68.7	50.0	1.4	57.1	66.1	60.1	66.2	60.2	54.8	62.4	67.6

Table 8: Results for the cluster- and topic-based UDA, where 0, 1, and 2 identify cluster/topic assignments given by the algorithm. The best score for each line is underlined, while bold indicates the best overall metrics.



(a) Baseline (b) UDA (c) CAT (d) CDCL

Figure 2: t-SNE visualizations of the feature representations for the BuzzFeed (source) and Hyperpartisan-L (target) datasets. Blue – source (src) mainstream, orange – target (trg) mainstreams, green – source hyperpartisan, and red – target hyperpartisan. Best viewed in color.



(a) K-Means (b) K-Medoids (c) LDA (d) NMF

Figure 3: t-SNE visualizations of the feature representations when employing topic/clustering methods on the validation sets. Blue – source (src) mainstream, orange – target (trg) mainstreams, green – source hyperpartisan, and red – target hyperpartisan. Best viewed in color.

UDA, the CAT, and the CDCL. Using different approaches to domain adaptation may reduce the domain gap in the feature space between the two domains. Still, many examples cluster together far apart from their counterparts. UDA obtains better representations than the other methods. When considering the topic-based adaptation (see Figure 3), we notice a better separation when employing topic models. Also, we achieve poor separation among classes for K-Means and K-Medoids.

6 Conclusions

In this work, we addressed the problem of transferring knowledge from fake to hyperpartisan news detection. We employed three types of architectures based on unsupervised training. We conducted multiple experiments, showing the effects of the hyperparameters in the given configuration. All employed methods manage to perform some domain adaptation. In particular, we showed that CDCL obtains the best results after applying data augmentation based on TF-IDF word replacement. In contrast, CAT managed the poorest results. By analyzing the t-SNE visualization, this model did not learn a good feature representation, with a minimal domain gap between the source and target datasets. The low accuracy we hypothesize is due to a lack of data from the source domain, as we have seen that data augmentation helped. For future work, we aim to investigate our approaches on other fake news datasets.

Acknowledgments

This research has been funded by the University Politehnica of Bucharest through the PubArt program.

References

Hadeer Ahmed, Issa Traore, and Sherif Saad. 2017. Detection of online fake news using n-gram analysis and machine learning techniques. In *Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments: First International Conference, ISDDC 2017, Vancouver, BC, Canada, October 26-28, 2017, Proceedings 1*, pages 127–138. Springer.

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2020. Do not have enough data? deep learning to the rescue! In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7383–7390.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. 2013. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Anshika Choudhary and Anuja Arora. 2021. Linguistic feature based learning model for fake news detection and classification. *Expert Systems with Applications*, 169:114171.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

Zhijie Deng, Yucen Luo, and Jun Zhu. 2019. Cluster alignment with a teacher for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9944–9953.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Chris Fraley and Adrian E Raftery. 2002. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American statistical Association*, 97(458):611–631.

Markus Freitag and Yaser Al-Onaizan. 2017. Beam search strategies for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 56–60, Vancouver. Association for Computational Linguistics.

Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030.

Abdullah Hamid, Nasrullah Shiekh, Naina Said, Kashif Ahmad, Asma Gul, Laiq Hassan, and Ala Al-Fuqaha. 2020. Fake news detection in social media using graph neural networks and nlp techniques: A covid-19 use-case. *arXiv preprint arXiv:2012.07517*.

Kazuaki Hanawa, Shota Sasaki, Hiroki Ouchi, Jun Suzuki, and Kentaro Inui. 2019. The sally smedley hyperpartisan news detector at SemEval-2019 task 4. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 1057–1061, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Thomas Hofmann. 1999. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, UAI'99, page 289–296, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Gerald Ki Wei Huang and Jun Choi Lee. 2019. Hyperpartisan news and articles detection using bert and elmo. In *2019 International Conference on Computer and Drone Applications (IConDA)*, pages 29–32. IEEE.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr.

Ye Jiang, Johann Petrak, Xingyi Song, Kalina Bontcheva, and Diana Maynard. 2019. Team bertha von suttner at SemEval-2019 task 4: Hyperpartisan news detection using ELMo sentence representation convolutional network. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 840–844, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Heejung Jwa, Dongsuk Oh, Kinam Park, Jang Mook Kang, and Heuiseok Lim. 2019. exbake: Automatic fake news detection model based on bidirectional encoder representations from transformers (bert). *Applied Sciences*, 9(19):4062.

Rohit Kumar Kaliyar, Anurag Goswami, and Pratik Narang. 2021. Fakebert: Fake news detection in social media with a bert-based deep learning approach. *Multimedia tools and applications*, 80(8):11765–11788.

Leonard Kaufmann. 1987. Clustering by means of medoids. In *Proc. Statistical Data Analysis Based on the L1 Norm Conference, Neuchatel, 1987*, pages 405–416.

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*, pages 829–839, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Boshko Koloski, Timen Stepišnik Perdih, Marko Robnik-Šikonja, Senja Pollak, and Blaž Škrlj. 2022. Knowledge graph informed fake news classification via heterogeneous representation ensembles. *Neurocomputing*, 496:208–226.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. In *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*, pages 18–26.

Daniel D Lee and H Sebastian Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

S. Lloyd. 1982. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore. Association for Computational Linguistics.

Federico Monti, Fabrizio Frasca, Davide Eynard, Damon Mannion, and Michael M Bronstein. 2019. Fake news detection on social media using geometric deep learning. *arXiv preprint arXiv:1902.06673*.

Ahmadreza Mosallanezhad, Mansooreh Karami, Kai Shu, Michelle V Mancenido, and Huan Liu. 2022. Domain adaptive fake news detection via reinforcement learning. In *Proceedings of the ACM Web Conference 2022*, pages 3632–3640.

Mihai Alexandru Niculescu, Stefan Ruseti, and Mihai Dascalu. 2022. Rosummary: Control tokens for romanian news summarization. *Algorithms*, 15(12):472.

Cristian Onose, Claudiu-Marcel Nedelcu, Dumitru-Clementin Cercel, and Stefan Trausan-Matu. 2019. A hierarchical attention network for bots and gender profiling. In *CLEF (Working Notes)*.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Andrei Paraschiv, George-Eduard Zaharia, Dumitru-Clementin Cercel, and Mihai Dascalu. 2021. Graph convolutional networks applied to fakenews: corona virus and 5g conspiracy. *UPB Scientific Bulletin, Series C: Electrical Engineering*, 83(2):71–82.

Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2018. Automatic detection of fake news. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3391–3401.

Matthew E Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. 2019. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54.

Martin Potthast, Johannes Kiesel, Kevin Reinartz, Janek Bevendorff, and Benno Stein. 2018. A stylometric inquiry into hyperpartisan and fake news. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 231–240.

Martin Potthast, Sebastian Köpsel, Benno Stein, and Matthias Hagen. 2016. Clickbait detection. In *Advances in Information Retrieval: 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20–23, 2016. Proceedings 38*, pages 810–817. Springer.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Maria Rae. 2021. Hyperpartisan news: Rethinking the media for populist politics. *New Media & Society*, 23(5):1117–1132.

Julio CS Reis, André Correia, Fabrício Murai, Adriano Veloso, and Fabrício Benevenuto. 2019. Supervised learning for fake news detection. *IEEE Intelligent Systems*, 34(2):76–81.

Robert M Ross, David G Rand, and Gordon Pennycook. 2021. Beyond "fake news": Analytic thinking and the detection of false and hyperpartisan news headlines. *Judgment and Decision making*, 16(2):484–504.

Somya Ranjan Sahoo and Brij B Gupta. 2021. Multiple features based approach for automatic fake news detection on social networks using deep learning. *Applied Soft Computing*, 100:106983.

Gerard Salton, Anita Wong, and Chung-Shu Yang. 1975. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. 2020. Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. *Big Data*, 8(3):171–188.

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.

Kai Shu, Suhang Wang, and Huan Liu. 2019. Beyond news contents: The role of social context for fake news detection. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pages 312–320.

Vertika Srivastava, Ankita Gupta, Divya Prakash, Sudeep Kumar Sahoo, RR Rohit, and Yeon Hyang Kim. 2019. Vernon-fenwick at semeval-2019 task 4: hyperpartisan news detection using lexical and semantic features. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 1078–1082.

Mateusz Szczepański, Marek Pawlicki, Rafał Kozik, and Michał Choraś. 2021. New explainability method for bert-based model in fake news detection. *Scientific reports*, 11(1):1–13.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Rui Wang, Zuxuan Wu, Zejia Weng, Jingjing Chen, Guo-Jun Qi, and Yu-Gang Jiang. 2022. Cross-domain contrastive learning for unsupervised domain adaptation. *IEEE Transactions on Multimedia*.

Yaqing Wang, Fenglong Ma, Zhiwei Jin, Ye Yuan, Guangxu Xun, Kishlay Jha, Lu Su, and Jing Gao. 2018. Eann: Event adversarial neural networks for multi-modal fake news detection. In *Proceedings of the 24th acm sigkdd international conference on knowledge discovery & data mining*, pages 849–857.

Albert Webson and Ellie Pavlick. 2022. Do prompt-based models really understand the meaning of their prompts? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2300–2344.

Liang Wu and Huan Liu. 2018. Tracing fake-news footprints: Characterizing social media messages by how they propagate. In *Proceedings of the eleventh ACM international conference on Web Search and Data Mining*, pages 637–645.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33:6256–6268.

Tong Zhang, Di Wang, Huanhuan Chen, Zhiwei Zeng, Wei Guo, Chunyan Miao, and Lizhen Cui. 2020. Bdann: Bert-based domain adaptation neural network for multi-modal fake news detection. In *2020 international joint conference on neural networks (IJCNN)*, pages 1–8. IEEE.

Xichen Zhang and Ali A Ghorbani. 2020. An overview of online fake news: Characterization, detection, and discussion. *Information Processing & Management*, 57(2):102025.

Xinyi Zhou and Reza Zafarani. 2020. A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Computing Surveys (CSUR)*, 53(5):1–40.

A Appendix

A.1 Results for Text Augmentation Based on GPT-2

Observing the improvements obtained using TF-IDF augmentation, we consider text generation an alternative. Therefore, we employ the GPT-2 model (Radford et al., 2019) to conditionally generate new examples given the news types (i.e., left-wing, right-wing, and mainstream). We follow an approach similar to the LAMBADA method proposed by Anaby-Tavor et al. (2020). Therefore, we fine-tune the GPT-2 base model on the hyperpartisan Buzzfeed dataset to generate new samples. Inspired by other works (Brown et al., 2020; Liu et al., 2023; Niculescu et al., 2022), we build the pre-training dataset using, for each sample, the following prompt:

```
News type :  <LABEL>
Text :  <TEXT>
<|endoftext|>
```

where `<LABEL>` is *left*, *right*, or *mainstream*, `<TEXT>` is the news content, and `<|endoftext|>` is the end token of the text. Since we use a relatively small context during experiments (i.e., 128 tokens), we do not require the auto-regressive model to learn to generate long samples, but rather more variation within the generated samples. To achieve this, we split each text into sentences and group every three sentences into one example under the same label.

As suggested by Kumar et al. (2020), during data generation, we iterate over each sample from the training set and prompt the model with `News type: <LABEL> Text:` followed by the first T tokens from each sample. Because the model may generate text that is not correlated with the label (i.e., either the model ignores the prompt label (Webson and Pavlick, 2022), or there is not enough data for the model to learn a clear distinction), we use the RoBERTa baseline model fine-tuned on the Buzzfeed dataset to filter the samples, ignoring those that do not match the model's prediction.

Text generation quality depends on the decoding strategy; thus, we explore multiple approaches.

Greedy decoding. The most trivial and fastest way of synthesizing text is to consider the token with the highest probability. Albeit simple, it has the disadvantage of generating repetitive and missing higher probability words behind lower probability ones.

Beam search. Beam search (Freitag and Al-Onaizan, 2017) seeks to solve the low probability issue from the greedy decoding by choosing the highest probability sequence within a number of beams. This method generally yields to higher probability sequence than greedy decoding. During experiments, we set the number of beams to 5.

Top-k. Using the top-k decoding (Fan et al., 2018), we consider only the highest k next tokens from the probability distribution over possible next tokens. This simple yet effective method produces more human-like text than previous approaches. In our experiments, we consider $k = 30$ tokens.

Top-p nucleus sampling. Introduced by Holtzman et al. (2020), the top-p nucleus sampling is an extension over top-k. We choose the tokens from the smallest subset whose cumulative probability is at least p instead of choosing from the top k probabilities. For experiments, we set $p = 99\%$.

To generate more samples, we repeat the procedure while setting $T \in \{3, 5, 10\}$. The results are shown in Table 9. CDCL obtains the highest scores on the source and target datasets using top-p and greedy decoding, respectively. On the source dataset, the accuracy reaches 97.8% and the F1-score tops at 97.7%, while on the target dataset, the best accuracy is 64.4% and F1-score is 70.4%. Compared with the TF-IDF text augmentation, the GPT-2 augmentation produces a higher best F1-score by 1% on the target test set, and achieves lower scores on the source test set by 1%. In addition, we notice that the performance improves when adding more data, especially on the source dataset, where we see an average improvement of 0.6% and 0.8% for accuracy and F1-score, respectively. On average, greedy decoding improves the target F1-score (i.e., 68.0±1.5%) while the lowest average is obtained by top-p (i.e., 65.7±3.5%). We notice a small improvement in favor of top-p compared with top-k on the source domain, but the target domain does not benefit from it in our case.

Decoding Strategy	T	UDA				CAT				CDCL			
		Source		Target		Source		Target		Source		Target	
		Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)
Greedy decoding	3	96.0	95.6	62.5	68.3	96.6	95.9	63.7	65.9	97.2	97.2	**64.4**	**70.4**
	3/5	96.0	95.6	60.1	**69.2**	96.6	95.9	63.9	66.6	96.3	96.3	61.7	68.6
	3/5/10	96.3	96.1	55.4	67.3	95.0	94.1	63.2	66.5	97.2	97.2	62.6	68.8
Beam search	3	95.7	95.3	**63.4**	68.2	94.4	93.1	63.5	64.1	94.7	94.5	64.2	68.1
	3/5	95.7	95.3	57.1	68.4	94.4	93.1	64.2	63.4	96.6	96.6	61.5	68.0
	3/5/10	**97.8**	**97.7**	62.1	68.9	96.3	95.6	**64.4**	66.1	96.9	96.9	60.7	66.2
Top-k	3	94.7	94.2	62.9	65.4	96.3	96.2	62.6	66.7	96.0	95.9	61.6	68.3
	3/5	95.0	94.7	61.7	68.6	96.9	96.8	63.8	65.9	96.9	96.8	60.7	66.7
	3/5/10	96.3	96.0	60.0	68.5	**97.2**	**97.2**	63.6	**68.4**	96.6	96.5	61.7	69.0
Top-p	3	95.7	95.3	61.5	67.1	96.9	96.8	63.3	61.4	97.2	97.1	63.6	69.1
	3/5	95.0	94.7	62.1	67.1	96.3	96.1	62.6	62.9	**97.8**	**97.7**	62.3	68.1
	3/5/10	95.7	95.3	61.4	68.3	96.3	96.2	61.5	59.3	97.5	97.5	62.5	67.9

Table 9: Results for the text augmentation using GPT-2. The source is BuzzFeed and the target is Hyperpartisan-L.