Imperial College London

MEng Individual Project

Imperial College London

Department of Computing

Detecting Political Bias On Reddit With Transfer Learning

Author:
Fawaz Shah

Supervisor:
Dr Anandha Gopalan

Second Marker:
Dr Francesco Belardinelli

June 20, 2021

Abstract

Identifying political bias on social media provides useful insights for analysts who want to quantify political sentiment online. However, tackling this task with traditional supervised machine learning is incredibly difficult due to the lack of annotated data available, unlike related tasks such as bias detection in news.

In this project we explore using transfer learning to detect political bias on social media, with knowledge learnt from detecting bias in news content. Specifically we look at unsupervised domain adaptation - a class of techniques developed for when no labelled data exists for the target problem. We investigate 'direct' transfer learning, followed by a state-of-the-art domain-adaptive version of BERT called AdaptaBERT. We then propose an extension to AdaptaBERT involving an extra Next Sentence Prediction fine-tuning stage, and evaluate this on a political bias detection task and a named entity recognition (NER) task. We find this yields a 3.3% improvement in F1 score for the NER task.

Further work is needed to help transfer learning classifiers reach the same performance as standard supervised classifiers, with the hope that they can exceed performance in the future.

Acknowledgements

Firstly I would like to thank my supervisor Dr Anandha Gopalan for his continuous guidance, enthusiasm and support throughout the project, during what has been a tough year.

Secondly, I want to thank everyone who I've had the pleasure of being friends with in the last 4 years. I especially owe a great deal to Andy, Subhash and Inara for proof-reading my report, as well as to Arjun, Zay and the rest of the *Lit Times* squad for keeping me motivated through the pandemic.

Finally I want to thank my amazing family for their endless love and encouragement through my entire degree. I wouldn't be where I am today without them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Political bias is being increasingly scrutinised in modern-day news media. The last decade has seen increasing political polarisation of news sources, most notably during the 2016 and 2020 US elections [1], even though the majority of people prefer unbiased news [2]. As such, a big focus has been placed on detecting political bias within news content.

However, less attention has been paid to detecting the biases of readers and consumers of news. We term this *readership bias*. Examining readership bias can help analysts quantify sentiment for political parties and politicians among reader communities and the general populace. Social media is a prime example of a medium where readership bias can be explored - almost 70% of Americans now get their news from social media [3] - and examining bias in social media comments regarding recent news can help tackle further problems on social media such as filter bubbles and echo chambers.

Most media watchdogs still perform bias detection manually [4] [5], using teams of analysts. These analysts can often only rate several dozen articles a month [6], and may introduce their own biases into their ratings. Recent research has increasingly focused on tackling bias detection problems with machine learning and natural language processing [7] [8], which can be much faster and less susceptible to internal bias. However, while it is possible to obtain ground truth bias data for news sources from media watchdogs, obtaining annotations for individual social media comments is much more challenging.

Transfer learning is a modern field of machine learning research that takes models trained for one problem A, and 'transfers' their knowledge so they can be applied for a separate, more challenging problem B. Specifically, *unsupervised domain adaptation* techniques are well-suited for when there is a lack of annotated data available for problem B. In this project we seek to explore if unsupervised domain adaptation methods can be used to detect political bias in social media content, given the knowledge learnt from bias in news content. We target Reddit due to its unique features such as subreddits and upvotes/downvotes, and because a significant majority of Reddit users get their news from the site (more than both Facebook and Twitter) [3]. Our main objectives are:

- To survey existing methods for detecting political bias in the news
- To explore, and possibly extend, existing domain adaptation methods to improve the performance of political bias detection on social media

1.1 Contributions

The main contributions of this dissertation are as follows:

- **Detecting political bias in news content** - We survey existing machine learning models for detecting political bias within news content, including ensemble classifiers that have not been explored in prior literature. We assess performance using pre-computed features from previous research, as well as using models such as BERT that perform textual feature extraction themselves (Chapter 3).

- **Developing a cross-domain Reddit dataset** - We create a novel dataset consisting of both news articles and Reddit comments reacting to those articles, aimed at domain adaptation tasks. We discuss the design decisions and trade-offs involved in creating a high-quality dataset, and assess similarity between the domains in our dataset (Chapter 4).
- **Exploring unsupervised domain adaptation techniques** - We investigate unsupervised domain adaptation methods to detect political bias on Reddit, using knowledge learnt from detecting bias in news content. We explore a direct transfer approach, and a state-of-the-art domain-adaptive BERT model called AdaptaBERT (Chapter 5).
- **Extending AdaptaBERT with Next Sentence Prediction** - We extend AdaptaBERT by adding Next Sentence Prediction to its fine-tuning stages, and evaluate its performance on a political bias detection task as well as a named entity recognition (NER) task. We do not see an improvement over standard AdaptaBERT for the political bias detection task, however we see a 3.3% improvement in F1 score for the NER task (Chapter 6).

Chapter 2

Background

In this chapter we cover several elements of background research needed for the project. We introduce the idea of political bias (Section 2.1), and look at existing datasets of news content annotated by political bias (Section 2.2), followed by machine learning methods that have been employed to detect bias in news (Section 2.3). We then introduce Reddit (Section 2.4) and the idea of readership bias (Section 2.5), and explore related work in detecting political readership bias on social media. Finally, we introduce transfer learning (Section 2.6) and specifically previous work in domain adaptation (Section 2.6.1).

2.1 Political bias

Discussions on political bias are mainly focused on whether a particular news source is left-wing or right-wing. The definition of each is not clear-cut, and the boundaries between what constitutes left-wing thought and right-wing thought varies across countries and across time periods.

A YouGov survey [9] in 2019 found that issues in the UK that matter most to left-wing voters include remaining in the EU, opposing the use of nuclear weapons, increasing the minimum wage, supporting industry nationalisation and increasing wealth tax, whereas right-wing voters support the opposite viewpoints. In the USA [10] left-wing viewpoints include expanding government healthcare programs and social security, tightening environmental regulation, increasing corporation tax, expanding amnesty to undocumented immigrants, while again right-wing viewpoints are in opposition.

Overarching themes that can be drawn include that left-wing thought more often than not supports expanding the role of government, whereas right-wing thought advocates for smaller government. Economically, left wing voters tend to support tight financial regulation, however right wing voters support less regulation in favour of a free market economy, and left wing voters often support globalisation and stronger international ties with other countries whereas right wing voters are more sceptical of globalisation and may prefer protectionism.

Despite these differences, political bias in text can be extremely nuanced, hence the interest in using automated natural language processing methods for bias detection.

2.2 Existing news datasets

In this section we look at some existing datasets of news sources and news articles that have been annotated by political bias.

2.2.1 Media Bias/Fact Check

Media Bias/Fact Check (MBFC) provides a list of over 3500 news sources, annotated with their respective political bias leanings [4]. The news sources covered are mostly US-based, and include mainstream and non-mainstream media - examples include CNN, Fox News, Breitbart, Bloomberg, and many more.

MBFC has a team of volunteers who manually assess and assign scores per-source based on content from their articles. Each source is scored from 1-10 on the four following categories: biased wording/headlines, factuality/well-sourced information, story portrayal, and political affiliation [11]. The scoring mechanism is shown in Table 2.1.

0-2	2-5	5-8	8-10
Least Biased	Slightly Biased	Moderately Biased	Extremely Biased

Table 2.1: MBFC scoring mechanism

The scores across the 4 categories are averaged to produce an overall score for each news source. Note that this scale only measures the degree of bias and not the direction of bias (e.g. left/right wing) - this is manually classified by MBFC's volunteers. MBFC also produces a detailed report giving more information about each source's particular traits, for example the following is a sample from the report on CNN:

> *"...we rate CNN left biased based on editorial positions that consistently favors the left, while straight news reporting falls left-center through bias by omission. We also rate them Mixed for factual reporting due to several failed fact checks by TV hosts. However, news reporting on the website tends to be properly sourced with minimal failed fact checks."* [12]

Note since most catalogued news sources are based in the USA, this scale is centred heavily on the US political scale.

2.2.2 News-Media-Reliability

Baly et al., as part of their analysis on detecting political bias [8], created the News-Media-Reliability dataset [13], a collection of engineered features from 1066 news sites, with political bias annotations provided by Media Bias/Fact Check. Sites are categorised according to 7 labels: *extreme-left, left, center-left, center, center-right, right* and *extreme-right*. The dataset features include textual features from each site's article content and its headlines, as well as information from the site's Twitter page and Wikipedia page (if they exist), and its Alexa web traffic rank. The full collection of features is given in Table 2.2.

Category	**Feature**	**Description**
Site traffic	`alexa`	Alexa ranking
URL	`url_structure`	Site URL
Articles	`articles_title_glove`	Article headline features
	`articles_body_glove`	Article body features
Twitter	`has_twitter`	Whether site has Twitter or not
	`twitter_created_at`	Year Twitter account was created
	`twitter_description`	Twitter account biography
	`twitter_engagement`	Twitter user engagement metrics
	`twitter_haslocation`	Whether site has location in their Twitter biography or not
	`twitter_urlmatch`	Whether Twitter account contains a link to the site URL
	`twitter_verified`	Whether Twitter account is verified or not
Wikipedia	`wikipedia_categories`	Wikipedia page categories features
	`wikipedia_content`	Wikipedia page content features
	`wikipedia_summary`	Wikipedia page summary features
	`wikipedia_toc`	Wikipedia page table of contents features

Table 2.2: Engineered features in the News-Media-Reliability dataset

The `articles_title_glove` and `articles_body_glove` features are not actually the text content of the headline/body, but rather a set of 141-dimensional custom features relating to the structure

of the text, sentiment scores, language bias analysis and also complexity of the text, expressed as 141-vectors. More details are given in Baly et al. [8]. The Wikipedia features are word2vec word embeddings of the corresponding text (see Section 2.3.1 for more detail). Overall this feature set is fairly comprehensive, as it includes a well-rounded representation of a news source's article content plus content taken from the source's online presence.

2.3 Detecting political bias in the news

In this section we will look at examples of techniques that have been applied to detect political bias in the news with NLP.

2.3.1 Word embeddings

In order to run machine learning models on words, they must first be converted into feature vectors. Word embeddings are a representation of words in a finite-dimensional vector space, such that the word embeddings of words that are semantically similar are closer to each other in the vector space. Word embeddings can be learnt manually by training a neural network to generate them using an embedding layer, or pre-trained models can be used such as Google's word2vec [14], or GloVe [15].

Word embeddings by themselves have been used to analyse bias in text. Bolukbasi et al. [16] use the cosine similarity between averaged word embeddings of Google News articles to detect gender bias, and Gordon et al. [17] use the same idea on tweets to detect political bias. However, they are most often used as a pre-processing technique for machine learning models.

2.3.2 SVMs

Support vector machines (SVMs) are supervised models that can be used for both classification and regression. In the classification case, finding the decision boundary between classes in feature space is formulated as an optimisation problem, which the classifier approximates using gradient descent. We can change the function being used to model the decision boundary, called the kernel. A common kernel used instead of the standard linear kernel is the radial basis function (RBF) kernel. This is useful for modelling classes which occur in spherical clusters rather than with straight-line boundaries.

Baly et al. [8] trained an SVM classifier to detect political bias in the News-Media-Reliability dataset. They used an RBF kernel, and provided F1 and accuracy scores for classifiers trained on individual News-Media-Reliability features (see Table 2.2) as well as groups of features, compared to a 'majority' baseline that always predicts the majority class in the dataset (i.e. the modal class). They achieved a highest F1 score of 61.31% and accuracy of 68.86% after merging small classes. The code and the dataset for this study are open-source and available on GitHub, making it easy to reproduce the paper's results. The authors do not compare SVMs with any other classifiers, providing ample opportunity for future work.

2.3.3 Ensemble classifiers

Ensemble classifiers involve aggregating predictions from a group of classifiers, called *base learners*, in order to improve accuracy when compared to individual classifiers. In the classification case, aggregation usually involves taking a majority vote from the base learners' predictions. In order for ensemble learning to work, the base learners should ideally be uncorrelated (that is, there should not be any patterns between predictions of any two base learners for the same training samples).

Two common ensemble learning techniques are *bagging* and *boosting*. Bagging (Bootstrap Aggregating) involves bootstrapping say, T new training datasets from an original dataset, and training T independent base learners on each new dataset. During inference time, a majority vote of predictions from each base learner is used to form an overall prediction. A popular bagging technique is a random forest, which is a ensemble of decision trees.

Boosting involves training base learners serially on the same training set, instead of in parallel as with bagging. In Adaboost [18], the first boosting algorithm created, any mis-classified examples from one learner are upweighted for the next learner, meaning an ensemble of the base learners

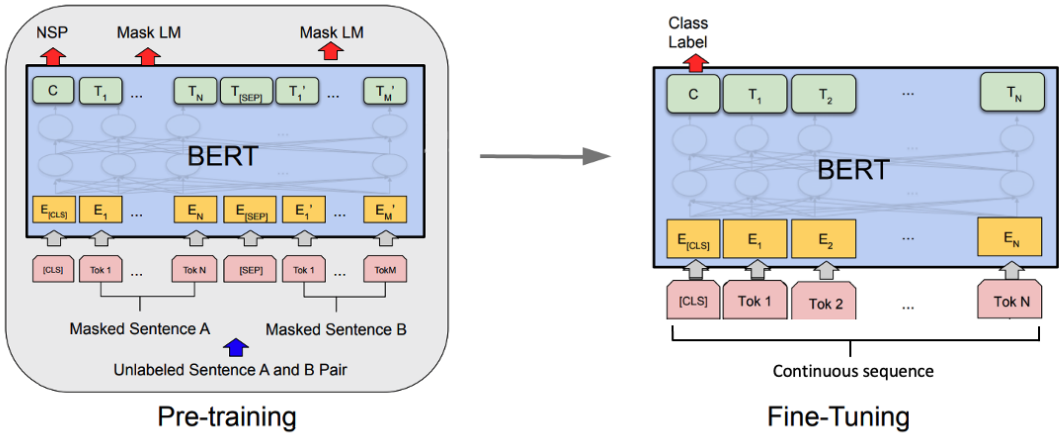


Figure 2.1: BERT pre-training and fine-tuning stages, in this case for sequence classification (edited from [21])

learns a good representation of all the data. Adaboost uses decision stumps as base learners (decision trees of depth 1). In a more modern technique called gradient-boosted trees [19], full decision trees are used as base learners, and residual errors from previous learners are factored in to each learner's training. Gradient-boosted trees can be bagged to form gradient-boosted forests.

Ensemble classifiers have not been extensively explored for the problem of detecting political bias. Voong et al. [20] compared gradient-boosted forests to Naive Bayes classifiers, SVMs, and logistic regression in detecting political bias of tweets, finding gradient-boosted forests gave the highest accuracy and F1 scores.

2.3.4 BERT

BERT (Bidirectional Encoder Representations from Transformers) [21] is a transformer-based model developed for natural language processing purposes. Transformer networks can learn contexts across extremely long sentences and paragraphs, and also enables bi-directional training that allows transformers to learn context even more deeply.

BERT is a stack of encoder models that utilises semi-supervised learning. This means BERT models are often pre-trained for use with a particular language, e.g. English, and then fine-tuned for a particular NLP task. Fine-tuning is often fast, as the main bulk of training is performed in pre-training. Pre-training involves two tasks:

- **Masked Language Modelling (MLM)** - random words in the training corpus are replaced with `[MASK]` tokens, and BERT is trained to predict the missing words. This helps BERT understand bi-directional context between words in a single sentence.
- **Next Sentence Prediction (NSP)** - BERT takes in two sentences A and B, and determines whether sentence B actually follows sentence A in the training corpus or not. This helps BERT understand context across lots of sentences.

A diagram of BERT during both pre-training and fine-tuning is shown in Figure 2.1. During pre-training, BERT generates its own contextualised word embeddings which it then uses throughout the encoder stack. BERT models come in varying sizes, the most popular being 'base' and 'large'. The base model contains 12 layers with each hidden layer containing 768 neurons, and gives 110M trainable parameters overall. The large model contains 24 layers each with 1024 neurons, and gives 340M trainable parameters. The large model often produces better results than base for popular benchmarks in sequence classification, question answering and other common NLP tasks, however it requires much more memory to train.

Input sequences into BERT must first have special `[CLS]` and `[SEP]` tokens added to them. A `[CLS]` token denotes the beginning of the input sequence, and `[SEP]` tokens are added between

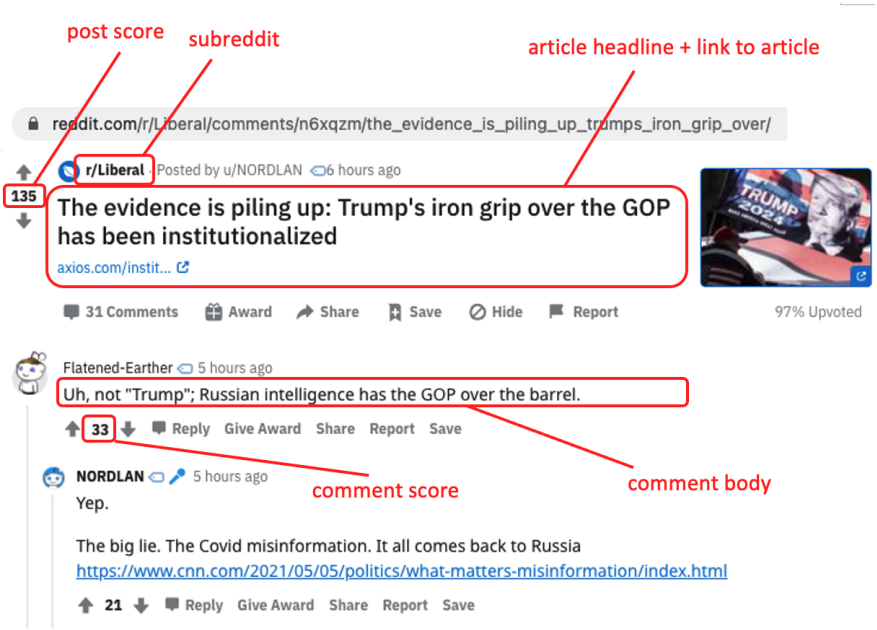


Figure 2.2: The anatomy of a typical Reddit post (image taken from r/Liberal [26])

sentences as markers for BERT. The input sequence is then converted to token IDs, segment IDs and attention IDs. For more information on these, see [21].

Chun et al. [22] used BERT to simultaneously detect political bias and detect trolls on Twitter, finding that the accuracy, precision and recall of the BERT model far surpassed those of the other models used (SVMs, standard neural networks and convolutional neural networks). They report a highest accuracy of 89% for the bias detection task. The Bipartisan Press in their analysis [7] found BERT performs much better at predicting both amount of bias and direction of bias than both LSTMs (long short-term memory networks) and ULMFiT (a transfer-learning-based model), as well as an ensemble model built from the two. They obtained even better results when using RoBERTa [23], a version of BERT that has been pre-trained for longer and with more data. Baly et al. [24] used BERT contextualised word embeddings extracted from article text as an input feature to an SVM trained to detect bias. They report a highest F1 of 81.27% and accuracy of 81.83%, both much higher than for their earlier SVM classifier without BERT.

2.4 Reddit

Reddit is one of the most popular social media platforms today, with over 430 million active users as of April 2020 [25]. Reddit is characterised by its unique 'subreddits' feature: communities that any user can join in order to read posts, comment, and post themselves about a particular topic. Subreddits begin with 'r/' followed by the topic name, for example popular subreddits include r/sports, r/astronomy, r/books and r/cooking. More politically-oriented subreddits include r/liberal, r/conservative, r/worldpolitics, r/ukpolitics, and many more. Reddit users can 'upvote' or 'downvote' any posts or comments, which increases/decreases their visibility to other users.

Figure 2.2 shows what a typical Reddit post looks like. The post and comment 'scores' are simply the sum of any upvotes and downvotes that particular post or comment has - an upvote counts as +1, a downvote as -1. Reddit provides a 'top posts' feature for each subreddit that allows users to search for the top posts by score in the last day/week/month/year.

A typical post on a political subreddit involves a link to a particular news article, and a comment stream with reactions from people who frequent that subreddit. Most often this will be the users that are actually subscribed to that subreddit, but occasionally there is discourse between subscribers and non-subscribers.

2.4.1 Collecting Reddit data

Reddit data can be collected from the public Reddit API [27], or from online data dumps of Reddit posts such as the PushShift dataset [28].

The Reddit API yields JSON files that describe any given subreddit, post, comment or user on the website. Examples of information exposed by the API include subreddit names, descriptions, post content, links to webpages referenced in posts, post scores, comment text and comment scores. The API also provides features to search for the most recent posts in a particular subreddit, and the top n posts in that subreddit in the last day/week/month/year. The API is most commonly accessed with the Python `praw` package [29].

The PushShift dataset is a large data dump of Reddit posts, comments, subreddit information and more, assembled by Reddit user Jason Baumgartner. It is available online at `https://files.pushshift.io/reddit/` [28]. The dataset contains Reddit posts and Reddit comments arranged by month from June 2005 (the month Reddit was created) through to December 2019 - in total the dataset contains around 651 million posts and 5.6 billion comments. The dataset contains almost all the same information the Reddit API exposes, however one cannot fetch the top n posts in that subreddit at a particular point in time.

In terms of data annotated by political bias, there is a severe lack of published datasets of Reddit content in this format. Almost all papers exploring political bias on Reddit (see Section 2.5.1) do not make their data available publicly.

2.5 Readership bias

We define readership bias as the bias shown by audiences, rather than by a news source. A significant portion of readership bias research so far has focused on analysing the sentiment of readers - early work on this was done by Lin et al. [30], who classified news articles by 8 different possible emotions expressed by the reader, recorded as reactions on Yahoo's website. Psychologist Paul Ekman popularised the existence of 6 basic emotions [31]: happiness, sadness, anger, fear, disgust and surprise. Strapparava et al. [32] used this emotion model to detect sentiment analysis in news headlines with a Naive Bayes classifier. Tan [33] also created a readership bias model based on the 6 emotions, and separately created a model to predict the distribution of Facebook Reactions on posts containing links to news articles.

2.5.1 Detecting political readership bias on social media

Social media is increasingly becoming a hub for political discussion and discourse. Garimella & Weber [34] analysed the political polarisation of Twitter discussions over time using retweets and hashtags used by 670,000 Twitter users, finding that polarisation increased by around 10-20% between 2009 and 2016. Rao et al. [35] used SVMs trained on n-gram features extracted from tweet text to predict the political affiliation of various Twitter users. Voong et al. [20] compared Naive Bayes classifiers, SVMs, logistic regression, and random forests with gradient boosting in predicting political polarity of tweets, also using an n-gram approach, finding logistic regression and random forests to give the highest accuracy and F1 scores, and Naive Bayes to give the worst.

Baly et al. [24] interestingly used features taken from readers' Facebook and Twitter profiles as a predictor for a news source's political bias. Baly's model of thinking was that a source's audience characterises how biased that source is and in what direction (that is, readership bias determines bias in the source). This differs from the mainstream line of thinking, which is that bias in the source determines readership bias. It is hard to say which of these philosophies is 'correct' - indeed there may be an element of both at play, that is news organisations will decide what to publish based on their target audience, and at the same time their target audience will change depending on what they publish.

Reddit has been less explored in this area compared to Facebook and Twitter. Kane & Luo [36] carried out an exploration into political communities on Reddit, examining whether non-political subreddits still exhibit some unconscious political leaning. They did this by analysing comments across different posts within these subreddits and using a Latent Dirichlet Allocation (LDA) model to see which topics appear most frequently in each subreddit. This topic model was then used to

train a classifier to predict the political bias of a subreddit, based on its most-common topics. They were able to achieve a highest accuracy of 85.2% with an SVM classifier using n-gram features. Since LDA is an unsupervised method, Kane & Luo did not have to source any Reddit data annotated by political bias, which is very scarce.

Predicting support for specific political figures on social media is a popular task, especially during widely-publicised elections such as the 2016 and 2020 US elections. Massachs et al. [37] predict support for Donald Trump using a combination of homophily and social feedback exhibited on r/The_Donald, a popular subreddit centred around support for Donald Trump. They report a highest accuracy of 35.5%, suggesting this problem is significantly more challenging than simply detecting left/right-wing bias. It is worth noting that the authors use participation in r/The_Donald to annotate Reddit users as Trump supporters or not, suggesting they assume only Donald Trump supporters will post in r/The_Donald. We examine this assumption in Chapter 4.

2.6 Transfer learning

Transfer learning is the study of training a machine learning model for a particular problem A, and applying the model to a different but related problem B. Transfer learning may be explored when it is hard obtaining suitable annotated training data for problem B, or because training for problem B would take too long otherwise.

One popular example of transfer learning is in BERT models - BERT is pre-trained on text from BookCorpus and English Wikipedia [21] for the tasks of masked language modelling and next sentence prediction, and is later fine-tuned for the target task at hand (which could be sentence classification, question answering, named entity recognition, etc.). Another popular form of transfer learning occurs in imaging problems, where convolutional neural networks pre-trained for image classification on ImageNet are then fine-tuned for other tasks such as object detection, image segmentation, or perhaps for further image classification.

In the following, we use notation similar to previous literature [38] [39] to formalise the theory of transfer learning:

A domain $\mathcal{D}$ can be formulated as a feature space $\mathcal{X}$ accompanied by a probability distribution $P(X)$ over the feature space. Intuitively, a domain is a set of data that exhibits a unique set of characteristics with respect to some feature space. In the real world, different textual domains include literary writing, scientific writing, news article text, tweets, etc. These will all exhibit a different distribution of features if we use a feature space of, say, n-dimensional word embeddings.

Given some domain $\mathcal{D} = (\mathcal{X}, P(X))$, a task $\mathcal{T}$ is defined as a set of labels $\mathcal{Y}$ and a prior (or 'ground truth') label distribution $P(Y)$. A classifier aims to learn the conditional probability distribution $P(Y|X)$ from some training data $\{x_i, y_i\}_{i=1}^{n}$.

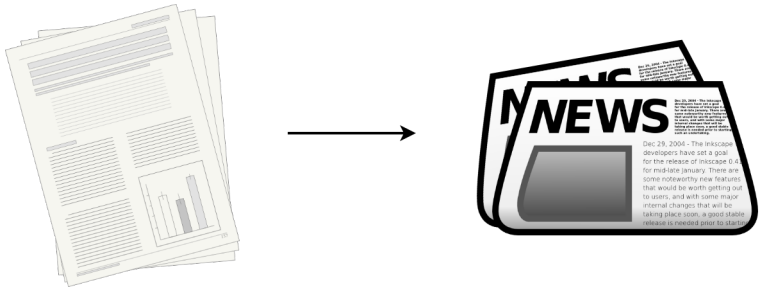
Given a **source domain** $\mathcal{D}_S$ and **source task** $\mathcal{T}_S$, along with **target domain** $\mathcal{D}_T$ and **target task** $\mathcal{T}_T$, the goal of transfer learning is to learn the conditional distribution $P_T(Y_T|X_T)$ in domain $\mathcal{D}_T$.

Transductive transfer learning is when the target task is the same as the source task (i.e. $\mathcal{T}_S = \mathcal{T}_T$), however the source domain and target domain may vary, or labelled data may only be available in the source domain. Examples of transductive transfer learning include domain adaptation and cross-lingual learning. *Inductive transfer learning* is when the two tasks are not the same, and labelled data for the target domain is often present. Examples of inductive transfer learning include multi-task learning and sequential transfer learning.

2.6.1 Domain adaptation

Domain adaptation (also known as cross-domain learning) is a specific case of transductive transfer learning. In this case the source task and target task are identical, however the source and target domains may be different, or labelled target data may not be available (in which case we perform *unsupervised* domain adaptation). Figure 2.3 shows a typical domain adaptation problem.

Daumé & Marcu [41] initially proposed the idea of 'in-domain' and 'out-of-domain' data, and applied these ideas to statistical learning theory. Other early work by Blitzer et al. [42] introduced



Domain A
Scientific literature
Named entity recognition

Domain B
News articles
Named entity recognition

Figure 2.3: An example domain adaptation problem, transferring knowledge from the domain of scientific literature to the domain of news articles, for the task of named entity recognition. Images sourced from Clker [40].

the idea of structural correspondence learning, where "pivot features" are selected that occur frequently in both the source and target domain, and behave similarly in both. This correspondence can then help a classifier perform a task such as POS tagging on the target domain.

Pan et al. [43] proposed a spectral feature alignment algorithm that can group domain-specific words into clusters, based on the distances between the domain-independent words in the corpus. Domain-specific words are those that occur most frequently in one domain, whereas domain-independent words occur frequently in both domains. Michelbacher et al. [44] consider domain-{specific/independent} features as a whole, and propose the idea of unsupervised feature adaptation for situations where no labelled target domain data is available. A classifier is trained on source domain data, and any domain-independent features are recomputed in an unsupervised fashion to align themselves with the target domain data, before being passed to the classifier to infer on the target domain data. Any domain-specific features are ignored.

The above methods all rely on some manual selection of features that are "domain independent" i.e. occur similarly in both the source and target domain. However, recent work has aimed to remove this manual step in the transfer learning process. Myagmar et al. [45] explored using BERT for cross-domain sentiment classification, finding that fine-tuning BERT on the source domain and running inference on the target domain (a process known as *direct transfer*) beat several pre-existing benchmarks for the task, using significantly less training data and with much faster training times.

Further work has been done to extend BERT for unsupervised domain adaptation. Ye et al. [46] use BERT's layer weights as features in a form of feature adaptation, and Du et al. [47] explore adding a post-training domain-distinguishing task to BERT's fine-tuning stages, and injecting adversarial learning into BERT.

Han & Eisenstein propose AdaptaBERT [48], a novel unsupervised domain adaptation approach which adds an extra stage of masked language modelling on the target domain during fine-tuning. This gives a significant performance boost to the classifier, especially on words and phrases seen at inference time that were not seen in the source domain at training time ('out-of-vocabulary' terms). AdaptaBERT yields several percentage points higher accuracy than direct transfer methods for a historical English POS tagging task, with 30% increase in accuracy on out-of-vocabulary terms. AdaptaBERT also outperforms existing methods on a Wikipedia → Twitter transfer learning task.

Gururangan et al. [49] compared this approach, which they name task-adaptive pre-training, to a general domain-adaptive pre-training approach where MLM is performed on a large amount of generic target-domain text that is not necessarily aimed at the specific task at hand. They find a combination of both methods provides the highest classifier accuracy for transfer learning tasks between news articles, IMDB reviews and biomedical scientific papers.

2.7 Going forward

In this chapter we have defined political bias and looked at previous work in detecting political bias in the news with NLP and machine learning. We have discovered that ensemble classifiers are under-explored for this particular problem, which we examine further in Chapter 3, where we also directly compare existing methods of detecting bias. We have also defined the related problem of readership bias and explored prior work in detecting readership bias on social media.

We then covered transfer learning and domain adaptation techniques. It is rare to find social media content that has been annotated by political bias, due to the challenging nature of collecting annotations from real users, which is where the power of unsupervised domain adaptation comes in. Unsupervised domain adaptation methods don't need labels in the target domain, making them particularly suitable to the task of transferring between news articles and social media comments. However, for evaluation purposes we do require annotated social media data (see Chapter 4 for more on this).

Chapter 3

Detecting political bias in news content

In this chapter we survey existing models to detect political bias in the news, using the News-Media-Reliability dataset [13]. Baly et al. already provide an SVM classifier that uses the News-Media-Reliability pre-computed features to detect bias (see Section 2.3.2). We compare this to other popular classifiers used in literature, as well as ensemble classifiers that have not been explored previously in detail (Section 3.1). We then assemble our own dataset of raw article text, and explore using features learned automatically from the text with BERT (Section 3.2).

3.1 Using ensemble classifiers

The pre-computed features in the News-Media-Reliability dataset include feature vectors based off article text, Twitter pages, Wikipedia pages and web traffic (see Section 2.2.2 for more details about the dataset). In an ablation study, Baly et al. found that the features extracted from the actual article content provided the best classifier performance, however the Wikipedia, Twitter and web traffic features were not very useful for predicting bias. This intuitively makes sense, as not much extra information about bias can be derived from the Wikipedia or Twitter content of the news source. We therefore don't consider individual Twitter, Wikipedia and web traffic features in our results - we simply compare the performance using all pre-computed features vs only article text features.

3.1.1 Data pre-processing

The News-Media-Reliability features we use are already provided in numerical vector format, so no additional pre-processing needs to be done here. However, there is significant class imbalance among the 7 labels used in the data.

The original labels are *extreme-left, left, center-left, center, center-right, right* and *extreme-right*. We merge *extreme-left* and *extreme-right* into *left* and *right* respectively, due to *extreme-left* being very small in size. We also remove the *left-center* and *right-center* classes, since *right-center* is small in size, and these are ambiguous transitionary classes that can't necessarily be merged into *center* or *left/right*. Table 3.1 shows the class sizes before and after pre-processing.

extreme-left	left	left-center	center	right-center	right	extreme-right
21	168	209	263	92	157	156

left	center	right
189	263	313

Table 3.1: News-Media-Reliability original classes (top) vs. final classes used (bottom)

There is still some class imbalance between left and right, however the overall imbalance has been greatly reduced.

3.1.2 Evaluation

We compare six classifiers in total: SVMs, multi-layer perceptrons, decision trees, random forests, Adaboost and gradient-boosted forests. We choose these to give us a good split between 3 non-ensemble and 3 ensemble classifiers, and so that we cover ensembles that involve both bagging and boosting. Random forests are of course a bagged version of decision trees, and Adaboost and gradient-boosted forests both implement different gradient-boosting algorithms.

For each experiment we use 5-fold cross validation, and within each fold we use an 80:20 train/test split. All classifiers are implemented using the Python `scikit-learn` package [50], using the `GridSearchCV` method for hyperparameter tuning.

Results are shown in Table 3.2, given to 2 decimal places. We report macro-averaged F1 score, accuracy, mean absolute error (MAE) and macro-averaged mean absolute error (MAE^M). MAE^M is a weighted average of errors for each class, where the weights are inversely proportional to the size of each class i.e. smaller classes' errors are upweighted more. This metric is much more robust to class imbalance than MAE.

Classifier	**All features**				**Only text-based features**			
	Macro-F1	**Acc.**	**MAE**	**MAE^M**	**Macro-F1**	**Acc.**	**MAE**	**MAE^M**
SVM	**67.07**	**68.76**	0.46	0.50	64.93	66.27	0.45	0.48
MLP	54.90	60.26	0.56	0.66	58.17	61.57	0.52	0.59
Decision tree	48.28	49.80	0.67	0.70	49.03	49.80	0.70	0.71
Random forest	59.52	64.71	0.49	0.57	57.95	63.53	0.51	0.60
Adaboost	59.04	62.48	0.53	0.61	59.23	61.44	0.54	0.59
Grad-boosted forest	66.17	68.10	**0.44**	**0.48**	**65.85**	**67.45**	**0.44**	**0.48**

Table 3.2: Bias detection results using pre-computed features across different classifiers. Highest performing classifier for each metric is highlighted in **bold**.

We see that using all features, the highest performing classifiers is the SVM, with gradient-boosted forests coming in a close second. Only using the text-based features, gradient-boosted forests perform the best. Hence ensemble classifiers do show a small improvement ($\approx 1\%$) over the other, weaker learners for this particular problem.

Baly et al. report an SVM macro-F1 score of 61.31% and accuracy of 68.86%. Our SVM manages to achieve a similar accuracy, however our macro-F1 score is around 6 percentage points higher. This could be attributed to our differences in dealing with class imbalance - Baly et al. merge *center-left* and *center-right* into the *center* class, whereas we actually remove the transitionary classes altogether. Our higher F1 score could signify that keeping a distinct boundary between left-wing and right-wing content improves classifier performance, even if the amount of training data available is reduced. Merging *left-center* and *right-center* into the *center* class may confuse the classifier, causing it to mispredict.

We note for all classifiers except for MLPs, the accuracy and F1-scores are higher using all the features vs only text-based features, however the difference is within around 2 percentage points. This backs up the claims by Baly et al., who report that the non-textual features do not provide much additional information to the model.

The best hyperparameters for our SVM were RBF as the kernel function, degree of polynomial decision boundary = 3, $C = 100$ and $\gamma = 0.001$. The best gradient-boosted forest hyperparameters were: no. estimators = 100, learning rate = 0.6 and minimum samples in a leaf node = 5.

	article headline	article body	bias
0	On the Ground at the Inauguration: The Only Th...	Will Sennott\n\nWEDNESDAY, JANUARY 20, 2021, W...	left
1	Under President Biden, Will the Yankees Return...	Thurman Munson and Reggie Jackson in 1977 From...	left
2	Gun Rights Absolutists Celebrate Martin Luther...	Will Sennott\n\nMONDAY, JANUARY 18, 2021, RICH...	left
3	Thugs in Blue	THE BEAT GOES ON … AND ON\n\nOnce Again, Polic...	left
4	HELL YEAH! Sheriff Clark Publicly DISEMBOWELS ...	Al Sharpton always has had a couple screws loo...	right
...	...	...	...
1649	UK Educators Rank-and-File Safety Committee di...	The UK Educators Rank-and-File Safety Committe...	left
1650	Make It Sing	Before I lay into the Democrats for missed opp...	left
1651	Bill Maher: The SPIN Interview	If you care at all about democracy and the way...	left
1652	Stephan Jenkins on What Culture Truly Means	“When bad men combine, the good must associate...	left
1653	Emergence Is Interactive: A Jeff Bridges and S...	Jeff Bridges is an Academy Award-winning actor...	left

1654 rows × 3 columns

Figure 3.1: Self-assembled dataset of article headlines, article body text and political affiliation

3.2 Using BERT

Whereas the previous ensemble classifiers take in manually engineered features, BERT models learn the optimal features straight from the raw text. We therefore create our own dataset of news article text (including headline and article body text) to use with BERT. These were scraped from the web using an open source tool called `NewsScraper` [51], which is based off of the Python `newspaper` [52] package.

We scrape from all news sources present in the News-Media-Reliability dataset. After class merging from Section 3.1 we have 732 sources to consider. Many of these sources use front-end code formats NewsScraper cannot parse - NewsScraper only managed to scrape articles from 432 of these news sources. We collect a maximum of 5 articles per news source, giving us a total of 1654 news articles (note that not all news sites had 5 articles available to scrape). For each article we store the article headline, article body text, and the political bias label for this news source as given by Media Bias/Fact Check (*left*, *center* or *right*). A portion of the dataset is shown in Figure 3.1. The distribution of classes is shown in Table 3.3 - there are a significant amount more right-wing articles than center or left articles.

left	center	right
430	567	657

Table 3.3: Class distribution in self-assembled article text dataset

3.2.1 Data pre-processing

We pre-process the text by performing lowercasing, punctuation removal, stopword removal and lemmatisation. We opt to keep in exclamation marks and question marks, since BERT can learn features over these to help it detect bias. The Python NLTK [53] package is commonly used for stopword removal, however we find it removes words such as ‘yourselves’, ‘ourselves’, ‘wouldn’t’, ‘shouldn’t’ etc. which could perhaps also be useful for our model. We therefore perform our own more minimalist stopword removal, using mainly basic determiners and prepositions i.e. ‘I’, ‘you’, ‘in’, ‘at’. Lemmatisation is performed with NLTK’s WordNet Lemmatizer [54].

We encode the labels numerically as *center* := 0, *left* := 1, *right* := 2. We also shuffle the data so that articles from different news sources are spread out in the dataset. The pre-processed dataset is shown in Figure 3.2.

	article headline	article body	bias
0	bidens america one nation us versus them	president joe biden sworn 46th president janua...	2
1	how get covid19 vaccine miamidade broward	keep new time free support us local community ...	1
2	arm mob storm capitol building during electora...	day will go down infamy arm mob storm united s...	1
3	frontier ebook release january 2021	download month new release include late specia...	0
4	change date vaccine news angry cricket coach	clancy overell wendell hussey kick off another...	0
...	...	...	...
1649	legal liability loom orgs behind rally incite ...	legal liability loom orgs behind rally incite ...	1
1650	merck france pasteur institute end development...	covid19 pandemic underscore need our company o...	0
1651	anthony mackie responsibility message captain ...	anthony mackie clear not all say he new captai...	1
1652	union just get rare bit good news from supreme...	supreme court announce monday will not hear bl...	1
1653	florida new hq maga movement	former president donald trump remain iffy 2024...	2

1654 rows × 3 columns

Figure 3.2: Self-assembled article text dataset after pre-processing

3.2.2 BERT architecture

Our model is a BERT stack built for sequence classification i.e. with a linear layer at the end and softmax activation to assign a label to the input sequence. We use the pre-trained uncased, base model of BERT since the large model requires too much memory to train with available hardware. We include token IDs, segment IDs and attention masks extracted from the text in our model. A diagram of the model is shown in Figure 3.3.

BERT only accepts sequences of maximum 512 tokens in length, including the special `[CLS]` and `[SEP]` tokens. The maximum headline length with special tokens is 103 tokens, so headlines can be padded up to 512 tokens. However, the maximum article body length is 14,373 tokens, much longer than the limit, so we must truncate article body text to 512 tokens when passing them into BERT.

3.2.3 Experimental setup

We run the BERT sequence classification model using:

1. only headlines
2. only article body text
3. headlines and body text concatenated together i.e. "`[CLS]` {headline} `[SEP]` {body}"

Our hypothesis is that using article headlines to predict bias will give higher accuracy than using body text, since headlines distill the essence of the article into just a couple of lines. We also predict using a combined input formed with both headlines and body text will result in higher model accuracy, since more information is being presented to our model.

We use a 70:10:20 train/validation/test split, giving us a training set size of 1157 samples and test set size of 331 samples. We don't use k-fold cross validation with this model due to the small dataset size. With 5 folds each classifier would only be trained with 231 training samples and tested on 66 samples - the training and test sets may not be representative of the overall class distribution.

We use the `AdamW` [55] optimiser, with a learning rate of $2 \cdot 10^{-5}$ after hyperparameter tuning. We use a batch size of 10 - we found batches larger than this lead to validation accuracy dropping. We also clip gradient norms to 1 at each training step to prevent any exploding gradients.

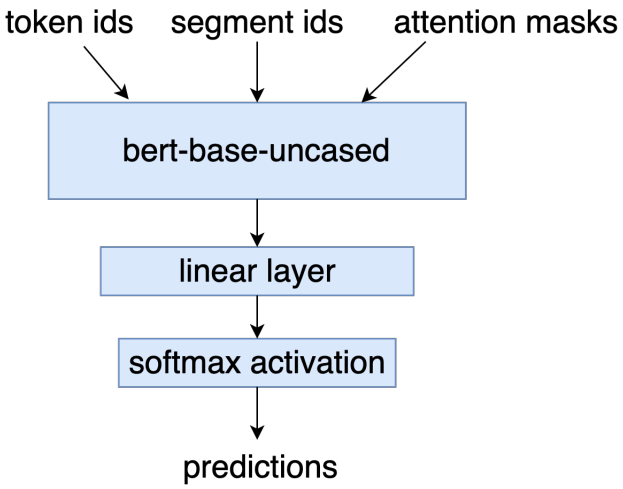


Figure 3.3: The BERT sequence classification model

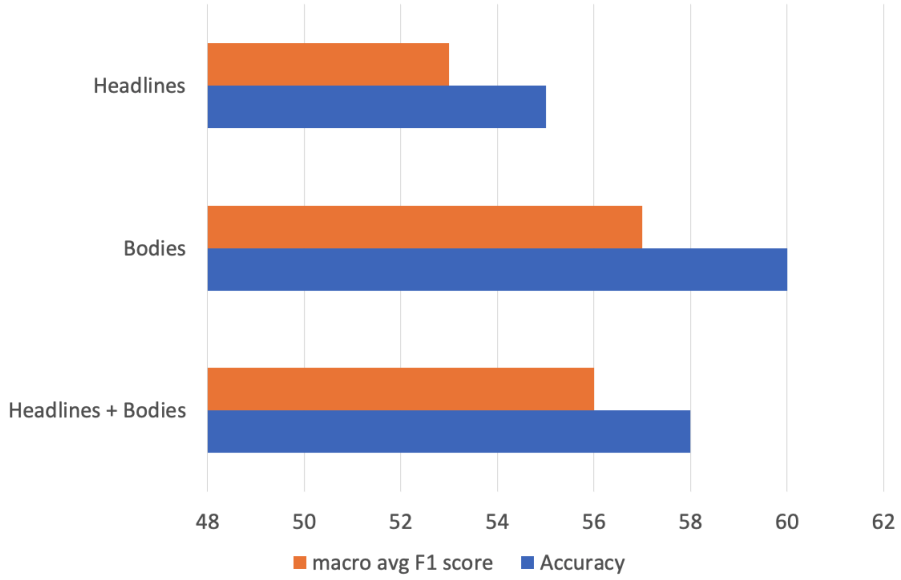


Figure 3.4: Evaluation metrics for BERT classifier applied to headlines and body text

3.2.4 Evaluation

Evaluation metrics comparing the 3 different input types described in the previous section are shown in Figure 3.4.

We can see using article bodies gives better performance than using article headlines by several percentage points in both accuracy and F1 score, with a highest accuracy of 60%. Headlines usually distil the content of the article into one or two lines, so this result could be attributed to article bodies simply containing more textual information than headlines (the longest headline in the dataset is only 103 tokens long, much shorter than the maximum 512 tokens allowed).

It's important to note that we have truncated the body text to just the first 512 tokens, and the beginnings of news articles often summarise their main points similar to how a headline would. This could also be why body text gives higher model accuracy than headlines. We explore this further in Section 3.2.5 by comparing using the beginning of the article to predict bias vs the middle or the end of the article.

Concatenating headline and body text performs better than simply using headlines, but worse than just using body text. This result is particularly interesting, as our initial hypothesis was that

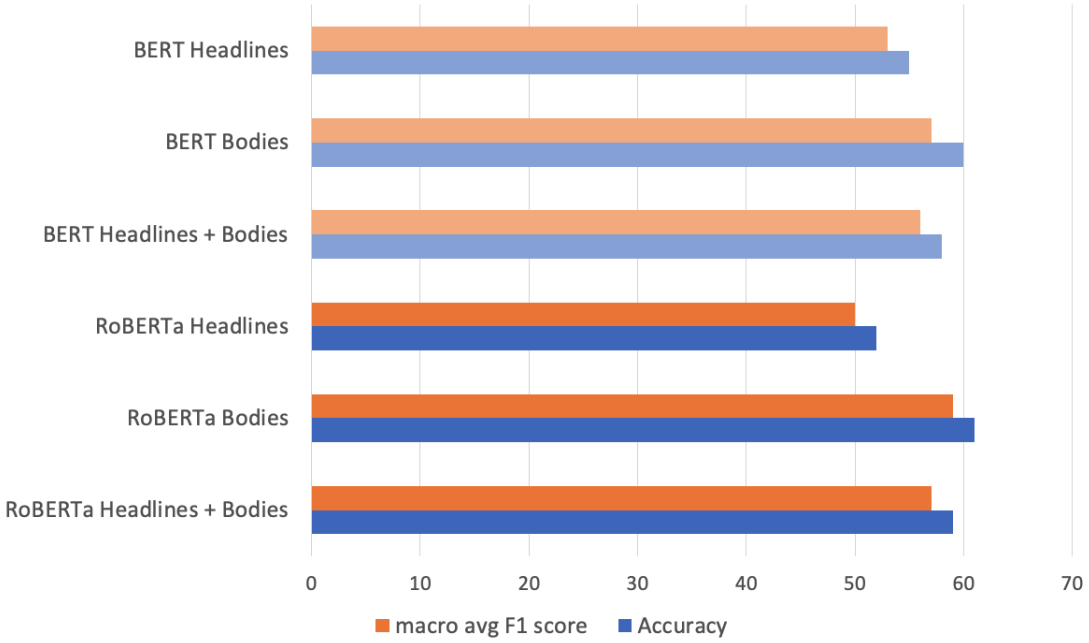


Figure 3.5: Evaluation metrics for RoBERTa classifier compared to BERT

unifying headline and body information would provide the model with more information than using either of the two input types individually. This, however, demonstrates that body text provides more information than any input involving the headlines.

We also explore using RoBERTa for sequence classification, a version of BERT that has been pre-trained further and with more data. Our results are compared in Figure 3.5. RoBERTa exhibits around the same performance as BERT, with a very minor improvement when using article body text only (61% accuracy). This demonstrates that further pre-training on the English language does not help the classifier detect bias more easily.

Chun et al. report an accuracy of 89% when using BERT to detect political bias in the Russian Troll dataset [56], a collection of tweets posted by Russian troll accounts. Our highest accuracy of 61% is much lower than this. One reason for this discrepancy could be availability of training data - the Russian Troll dataset contains around 3 million tweets, whereas our dataset is very small in comparison, consisting of only 1654 text samples.

Compared to our classifiers in Section 3.1, the performance of our BERT classifiers seems to be slightly worse. We achieve F1 scores in the range of 55-60% with BERT, whereas SVMs and gradient boosted trees were able to achieve F1 scores of 65-67%. This could be related to the feature set used in Section 3.1 - the News-Media-Reliability textual features are a complex set of features based off of the structure of the text, sentiment scores, language bias analysis and also complexity of the text. This feature set therefore provides the classifier much more information than what BERT can extract from raw text. However, the authors do not publish exactly how they computed their custom features, meaning future work on detecting bias with this feature set would be restricted to only the news sources they computed features for.

Building off the RoBERTa idea, further work could be done examining BERT models pre-trained via a political bias detection task, on top of the MLM and NSP tasks, to see if this improves bias detection. We could also explore applying BERT to a much larger dataset - while our self-assembled dataset contains high-quality raw article text, it contains a very small number of text samples compared to other datasets explored in previous literature.

One point to note is that BERT models split tokens up using a subword-level vocabulary, created using the Wordpiece [57] algorithm. Names therefore get split into subwords after tokenisation, e.g. “biden” becomes [‘bid’, ‘##en’] and “covid” becomes [‘co’, ‘##vid’]. This could limit the effectiveness of names within the BERT model, since the model will treat the ‘bid’ and ‘co’ tokens just like any other instances of those words in the text. We find that “biden” occurs 2,696 times

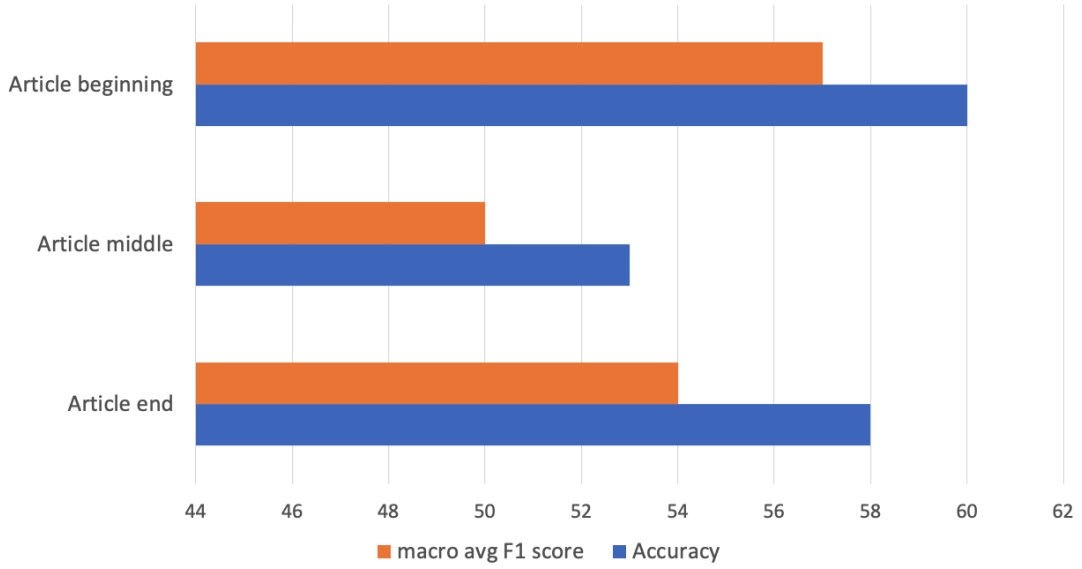


Figure 3.6: Evaluation metrics using beginning, middle or end of article text to predict bias

in the dataset, but the 'bid' token otherwise only occurs 11 times, so this is not a huge problem in this case. However, "covid" occurs 1,201 times in the dataset and the 'co' token otherwise occurs another 529 times, which could lead to confusion in the BERT model. Other weaknesses of Wordpiece tokenisation have been explored previously [58].

3.2.5 Detecting bias from particular article sections

In this section we purely focus on article body text. We want to examine which section of an article (beginning, middle or end) is most helpful to the classifier in determining political bias.

We use the same dataset as in the previous sections. Instead of letting the BERT tokenizer truncate articles to just the first 512 tokens, we isolate 512 tokens from the beginning, middle, and end of article bodies, and pass these into our BERT classification model. Our hypothesis is that using the beginning/end of article text will give higher model accuracy than using text in the middle, since articles often introduce or conclude their main ideas in these sections, and so evidence of political bias will be more evident.

We use the same experimental setup and hyperparameters as in the previous section. Results are shown in Figure 3.6.

We can see our hypothesis has been validated - the model accuracy is better when using text from the beginning or end of an article, rather than from middle sections. Interestingly using introductory text gives a higher accuracy than using concluding text - since introductory text is the first thing the reader sees, perhaps this text is more representative of the overall bias of the article.

The accuracies using the introduction and conclusion of the body text are 60% and 58% respectively, however the accuracy we achieved from using article headlines (in the previous section) is 55%. Therefore, using either the introduction or conclusion of the body text is actually more useful to the classifier than using any headline information at all.

Chapter 4

Developing a cross-domain Reddit dataset

In order to explore transfer learning between the domains of news articles and social media comments for the task of political bias detection, we create a novel cross-domain dataset containing a set of news articles that have been shared on Reddit, and the accompanying Reddit comments, both annotated by political bias. We create our own such dataset since, as mentioned in Section 2.4.1, no pre-existing datasets of this type exist in the literature.

In this chapter we discuss the challenges faced in creating a high-quality dataset, and the data sources we consider. News articles and Reddit comments present different (but related) domains of text that can be explored and compared - in this chapter we also examine similarities between the different domains in our dataset.

We mentioned in Section 2.6.1 that unsupervised domain adaptation methods do not need labelled data in the target domain, so we could theoretically explore transfer learning from news articles to social media comments without needing annotated comments. However, in order to evaluate performance of these domain adaptation methods with metrics such as F1 and accuracy, comments with labels are inevitably needed. Hence, in our dataset both news articles and Reddit comments are annotated by political bias.

4.1 Designing a high-quality dataset

Here we look at several issues faced in creating a high-quality dataset.

4.1.1 Finding suitable annotations

One major design decision we face is how to find suitable annotations for Reddit comments, since this influences later decisions on where (and how) to source our data.

Typically, posts and comments on social media are not annotated with political leaning, perhaps explaining why the task of exploring political bias on this medium is such a challenging one. However, we can exploit one of the unique features of Reddit, that of subreddits. Posts and comments on Reddit are naturally grouped into communities, and a wide range of popular and politically-related subreddits are available to browse on Reddit, such as r/liberal, r/conservative, r/republican, etc. These subreddit titles provide natural political-bias annotations for the content displayed within each of those subreddits.

One key assumption to examine here is that all content under a particular subreddit will necessarily follow that particular political leaning. For example, whereas most comments in r/liberal may be from subscribers to that subreddit, nothing stops members of another subreddit such as r/conservative from leaving comments under r/liberal and starting a discussion.

However, it has been shown that this occurs very infrequently. A study by Guimaraes et al. [59] quantified and examined cases of *harmony* and *dispute* among Reddit posts, where *dispute* is

defined as a comment thread where one particular set of comments attracts a notable amount of downvotes from the community, and *harmony* is a comment thread where no such set of downvoted comments exists. They find that in two major political subreddits (r/politics and r/worldpolitics) harmonious discourse makes up the majority of all comments, and disputes only occur in around 2% of comments. This reinforces the assumption we make that content within a particular subreddit will most likely follow that political leaning, since political conversations or arguments between users with opposing viewpoints will almost always contribute to dispute.

In cases where harmonious discourse may not make up a significant portion of all comments, we protect against this by selecting only posts and comments with high upvote counts. Content with more upvotes are more likely to follow the overall bias leaning of the subreddit, so selecting these will almost guarantee the subreddit-name annotations apply to that text. The selection process we use is shown in Algorithm 1.

Algorithm 1 Reddit dataset collection

```
subreddit list = [...]                                   ▷ See Section 4.1.2
for each subreddit do
    Assign subreddit a bias label
    Get top 300 posts this year with score > 10
    for each post do
        Store linked article (headline + article body)
        Get all comments with score > 10
        for each comment do
            Store comment text
        end for
    end for
end for
```

4.1.2 Selecting subreddits

Since our annotating process relies on subreddit titles, we must select subreddits that explicitly have a political leaning in their name. Examples include r/liberal, r/democrats, etc.

Our original plan was to keep the 3 labels used for our methods in Chapter 3: *left*, *center* and *right*. However, it is quite hard to identify accurate "centrist" subreddits, due to the vague definition of centrism. Furthermore, subreddits that by name may appear to be fairly centrist e.g. r/worldpolitics or r/politicaldiscussion, have been shown to be left-leaning due to Reddit's overall left-leaning bias [60]. We therefore restrict our classification problem to only 2 labels: *left* and *right*.

One simple way of collecting verifiably left/right-wing content is to explore subreddits that follow politicians themselves, e.g. r/obama. We look at collecting data from subreddits that follow recent US presidential election candidates, such as Barack Obama and Hillary Clinton. However, one problem is that many of the subreddits related to the 2016 and 2020 Republican Party candidate, Donald Trump, have been banned permanently from Reddit (such as r/The_Donald). We mitigate this issue by selecting content from r/sh*tliberalssay, a popular right-wing subreddit that has attracted many Trump supporters in recent years. The subreddit focuses on archiving "the worst liberals on Reddit" according to the subreddit description, which may prove to be an issue if the posts in the subreddit link to left-wing news articles, however the vast majority of link posts are actually links to images, and so don't get picked up by our scraper.

The full list of subreddits we collect data from is shown in Table 4.1.

4.1.3 Resolving class imbalance

One step we perform to improve the reliability of this dataset is to balance the *left* and *right* classes. Our dataset before balancing contains 992 Reddit posts (i.e. 992 news articles, since the posts are simply links to articles) and 50,608 comments. The class distributions in both articles and comments are shown in Table 4.2. We see that the majority class present in articles is *left*, but

Subreddit	Bias Label
r/liberal r/democrats r/obama r/hillaryclinton r/sandersforpresident	left
r/conservative r/republican r/sh*tliberalssay r/libertarian	right

Table 4.1: Scraped subreddits and their assigned bias annotations

the majority class in comments is *right*, presenting a challenge as to how to resolve class imbalance overall.

left	right
600	392

left	right
5,977	44,631

Table 4.2: Class distributions in news articles (top) and Reddit comments (bottom) before balancing

We can't resolve class imbalance in articles and comments independently, since we need to preserve data integrity by only keeping comments for which the articles they are reacting to are still present in the dataset, and vice versa. We must therefore balance either articles or comments, and then propagate changes through to the other domain by adding/deleting invalidated data. Thus we can only perfectly balance either articles or comments - the other domain will remain somewhat unbalanced in either case.

We first try balancing articles by downsampling the minority class and then removing all comments that correspond to deleted articles. However, this leads to catastrophic class imbalance in comments (approximately 44,000 *right* comments vs 2000 *left* comments). We therefore balance comments first through downsampling and then propagate changes to the articles - this leads to a much more manageable class imbalance in articles. The final class distributions are shown in Table 4.3.

left	right
419	387

left	right
5,977	5,977

Table 4.3: Class distributions in news articles (top) and Reddit comments (bottom) after balancing

Our final dataset therefore has 806 articles and 11,954 comments. This is a dramatic reduction in the number of comments, but is necessary to balance classes without upsampling.

The distribution of subreddits in our dataset is shown in Table 4.4. We can see most articles are sourced from r/liberal, r/conservative and r/libertarian. Furthermore, most comments are from r/liberal, r/sandersforpresident, r/conservative and r/libertarian.

4.1.4 Avoiding concept drift

Since we are storing cross-domain data for the purpose of comparing between textual domains, we must make sure to keep all other possible variables constant across all our data (e.g. time,

Bias Label	Subreddit	No. Articles	No. Comments
left	r/liberal	255	2,332
	r/democrats	27	646
	r/obama	33	42
	r/hillaryclinton	59	192
	r/sandersforpresident	45	2,765
right	r/conservative	157	3,202
	r/republican	37	155
	r/sh*tliberalssay	4	53
	r/libertarian	189	2,567
		806	**11,954**

Table 4.4: Subreddit distribution in our dataset

geography), so we don't inadvertently introduce other relationships into our data. This is known as "concept drift" [61].

We keep time domain the same across all data by only selecting posts from the current year (see Algorithm 1). We also keep geographical domain the same across all content by selecting only from mainly US-based subreddits (see Table 4.1). In fact this restriction does not matter too much, since over 70% of Reddit users are based in the USA [25].

4.2 Data sources

In collecting data from Reddit there are two main available sources we consider - the public Reddit API, or the PushShift Reddit dataset (see Section 2.4.1).

The benefits of using the PushShift dataset are that billions of posts and comments have already been scraped from Reddit, so less time and work is needed to manually scrape Reddit. The data is wide-ranging and covers thousands of subreddits, including the subreddits we selected in Section 4.1.2. However, Gaffney & Matias [62] have discovered several major warning signs about the PushShift dataset, most notably concerning large amounts of missing subreddit data, and corrupted data that doesn't make any sense (e.g. comments with timestamps earlier than the post itself).

On top of this, the main benefit of the Reddit API is that the top n posts of the last week/month/year can be collected, which is not provided in the PushShift data. This is ideal for our collection strategy as discussed in Section 4.1.1, since we want to collect posts and comments that have collected the most upvotes possible, which are grouped in the 'top' category as provided by Reddit. We therefore scrape all our data using the Reddit API.

4.3 Examining similarity between domains

In order to assess the viability of domain adaptation between the domains in our dataset, we look at the similarity between the domains. We model our dataset as containing 3 textual domains - article headlines, article body text, and Reddit comment text.

Jaccard distance is often used as a similarity metric comparing two sets. For any two sets of words A and B, the Jaccard distance between them is formulated as:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{4.1}$$

i.e. the number of words shared divided by the total number of words in both sets. Jaccard distance is always between 0 and 1 inclusive, where 0 indicates zero similarity (i.e. no words are shared between the two sets) and 1 indicates perfect similarity (i.e. the sets are identical).

We compare the Jaccard distances between vocabularies in each domain, which will show us how many words are shared between the domains. The Jaccard distances between each possible pairing of domains are shown in Table 4.5, to 3 s.f. We can see that the similarity between headlines and both other domains is very low. This is surprising - we would expect headlines and bodies to share a similar vocabulary since any article headline will of course have been derived from its corresponding body text. However, there is a significant similarity between article body text and Reddit comments, suggesting these domains are good candidates for domain adaptation. This is explored further in Chapter 5.

Domain Pairing	**Jaccard Distance**
article headlines - article bodies	0.0303
article headlines - comments	0.0352
article bodies - comments	0.443

Table 4.5: Jaccard distances between domains in our dataset

A Venn diagram showing overlap between vocabularies of the 3 domains is shown in Figure 4.1, with the size of each set labelled. We can see a large overlap of 159,472 words between article body and comment vocabularies, thus the high Jaccard distance between these 2 domains. However, there are still a large number of words in the article body and comment vocabularies that don't overlap with any other vocabularies (130,123 and 80,594 respectively).

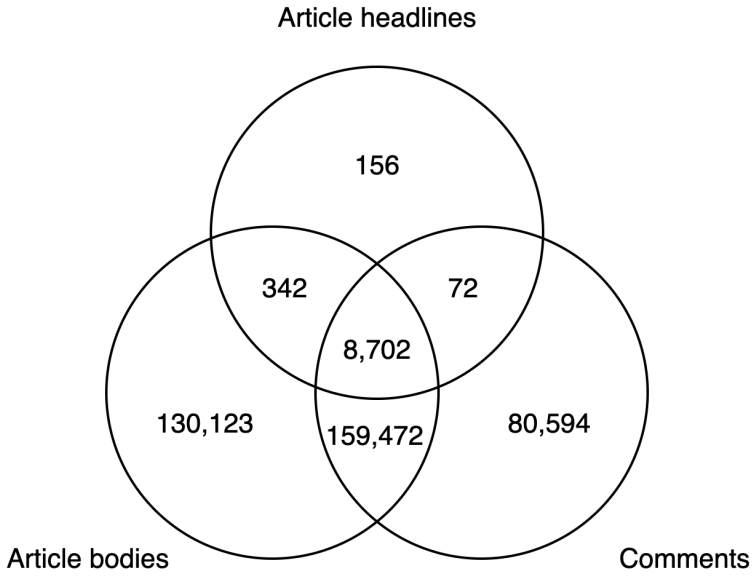


Figure 4.1: Venn diagram depicting overlap between each of the 3 domains in our Reddit dataset

Chapter 5

Exploring unsupervised domain adaptation techniques

In this chapter we explore unsupervised domain adaptation methods for detecting political bias in Reddit comments, using the cross-domain dataset developed in Chapter 4. We consider a direct transfer approach, and an approach using a state-of-the-art domain-adaptive BERT model called AdaptaBERT.

First we introduce some transfer learning notation, which we use throughout Chapters 5 and 6.

5.1 Notation

Let $A \rightarrow B$ denote some transfer from domain A to domain B, i.e. an $A \rightarrow B$ model is a transfer learning model from source domain A to target domain B. This will commonly involve training on some combination of domains A and B, and then making inferences on domain B.

We denote a standard classifier for domain B as an *in-domain* model for domain B. Even though this is not a transfer learning classifier, we say the source domain and target domain are both B. An in-domain model for domain B provides a theoretical upper-bound for the accuracy of an $A \rightarrow B$ model [48], since any $A \rightarrow B$ model will most-likely never be able to achieve the same accuracy as a model that has been trained solely on target domain B for inference on B.

5.2 Direct transfer

Direct transfer is the simplest form of domain adaptation. An $A \rightarrow B$ direct transfer model undergoes training solely on domain A, followed by inference on domain B and appropriate evaluation metrics being recorded. A diagram of the process is shown in Figure 5.1.

Using our cross-domain Reddit dataset, we will evaluate direct transfer from news articles to comments (denoted as *articles* $\rightarrow$ *comments*), and also transfer of comments to news articles (denoted as *comments* $\rightarrow$ *articles*). The latter is influenced by the discussion in Section 2.5.1 as to whether bias in the news source determines readership bias, or whether bias in the source's audience influences biases in the articles the source produces. Evaluating this transfer will help us see if audience reaction is a good predictor of news article bias. We compare these transfer models against in-domain models for articles and comments.

From Section 4.3, we identified 3 textual domains in our data. However, news articles can be split up into 2 domains - article headlines and article bodies. We choose to only use article bodies in our investigation, and ignore article headlines - from our results in Section 3.2.4, we found that using article body text to detect bias outperforms using any article headline information. From now on when we refer to 'articles', we are only considering article body text.

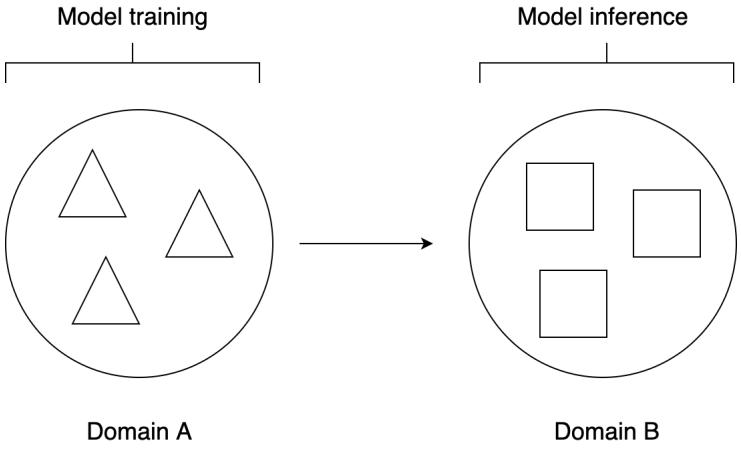


Figure 5.1: The direct transfer process (adapted from [38])

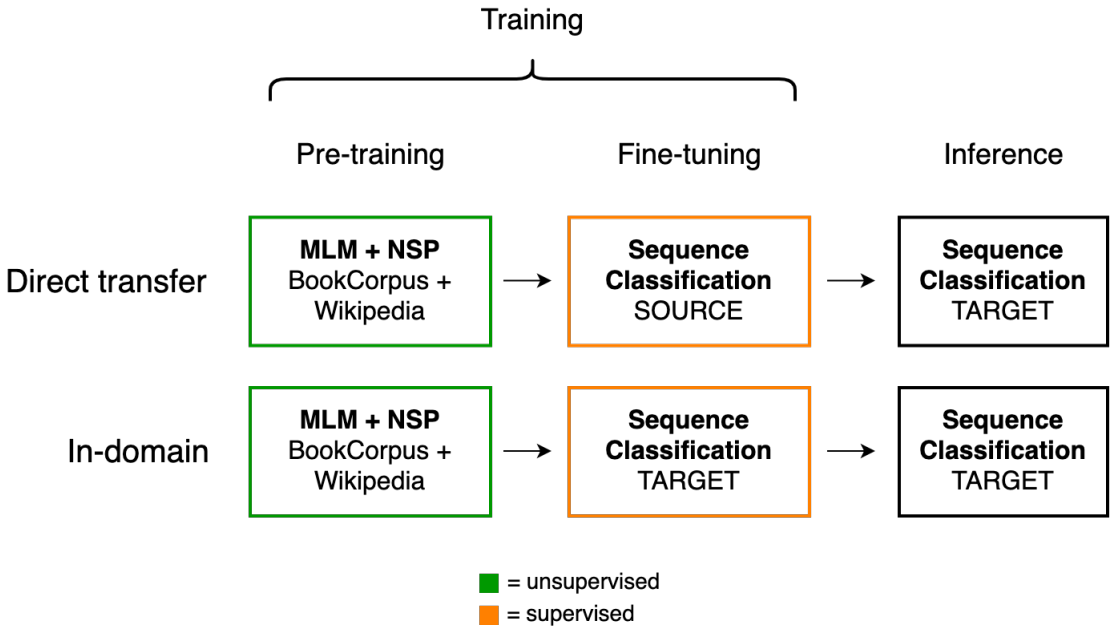


Figure 5.2: The stages of direct transfer and in-domain BERT approaches

5.2.1 Model choice

In choosing what model to use for our domain adaptation methods, we consider the models explored in Chapter 3. SVMs and gradient-boosted forests exhibited the best bias-detection performance out of all classifiers, however one major challenge is that these rely on the News-Media-Reliability custom features, which were solely computed for the news sources in that dataset. Recomputing these custom features for every news article and Reddit comment in our cross-domain dataset from Chapter 4 would be incredibly time-consuming, and so we choose BERT, which performs all feature extraction automatically via examining raw text input. We saw in Section 3.2 that BERT performs slightly worse than the News-Media-Reliability classifiers, however overall its performance is still fairly good.

We use BERT sequence classification models for all of the above. With BERT direct transfer models, pre-training is performed in the same way as for standard BERT - on a large amount of English language text sourced from BookCorpus and Wikipedia. Fine-tuning is performed on the source domain, and then inference is performed on the target domain. See Figure 5.2 for a diagram comparing direct transfer and in-domain BERT models. The training objective at each stage is shown in bold, with the domain being trained on underneath.

5.2.2 Experimental setup and evaluation

We use a BERT model for sequence classification based on `bert-base-uncased`, and perform text preprocessing in a similar fashion to our experiments in Section 3.2. We also use a train/validation/test split of 70:10:20 to match our Section 3.2 experiments. After tuning hyperparameters we achieve highest accuracies with 5 epochs of training and a batch size of 10.

We evaluate *articles* → *comments* direct transfer against an in-domain comments model, and *comments* → *articles* direct transfer against an in-domain articles model. We report macro-averaged F1 score and accuracy for all 4 models. Results are shown in Table 5.1.

Note that the metrics for in-domain models were collected from test-set inference, whereas metrics for the domain adaptation methods were collected from inference over the entire target domain. This is true of all the results we present in Chapters 5 and 6.

Source Domain	Target Domain	Model Type	Macro-F1	Accuracy
articles	comments	direct transfer	54.4	51.8
comments		in-domain	67.2	67.6
comments	articles	direct transfer	69.6	69.3
articles		in-domain	76.2	76.5

Table 5.1: Evaluation metrics comparing direct transfer to non-transfer baselines

We can see the direct transfer approaches are beaten by the in-domain classifiers in both F1 score and accuracy. The F1 score difference is around 13% for *articles* → *comments* transfer, and around 7% for *comments* → *articles* transfer. This suggests direct transfer by itself cannot completely substitute for training on the target domain, irrespective of which domain is the target.

The *comments* → *articles* transfer performs significantly better than *articles* → *comments*, both in terms of absolute metrics and in terms of how close to the in-domain baseline it is able to achieve. This suggests there is merit to the idea that looking at comments on a particular article are a good predictor for the bias of the article itself. This could be because the number of comments in our dataset is much larger than the number of articles (each article in our dataset has 14 comments on average), so the classifier has access to much more data to train on.

To test this idea, we experiment with reducing the number of comments in the dataset, and re-running the *comments* → *articles* transfer model. Reducing the number of comments by 50% results in an accuracy drop of around 5%, from 69.3% to 64.5%, and reducing by 75% gives a further accuracy drop of 15%. These are significant drops in accuracy, indicating the sheer number of comments in our dataset contributes to the high accuracy of transfer.

This is an important finding, since in the real world comments are much more abundant in nature than news articles - on articles from mainstream news outlets it is common to find hundreds of comments, both on the site itself and on social media such as Facebook and Reddit. The fact that comment text is a good predictor for news article bias with only a fairly small deterioration in accuracy compared to in-domain methods is something that can be explored further in future work.

5.3 AdaptaBERT

Beyond direct transfer BERT models, we explore a more modern method called AdaptaBERT [48], a modified BERT architecture that is geared towards unsupervised domain adaptation.

Standard BERT models are trained for masked language modelling (MLM) and next sentence prediction (NSP) during pre-training, and then are fine-tuned for the task at hand (in our case, sequence classification). In the direct transfer case, this fine-tuning occurs only on the source domain. AdaptaBERT adds an extra fine-tuning step where another round of MLM is performed, but on a combination of the source domain and the target domain. This helps attune BERT's internal contextualised word embeddings to words in the target domain, and yields significantly better performance than direct transfer for the tasks of POS tagging and named entity recognition

[48], especially on 'out-of-vocabulary' (OOV) terms - words seen at inference time in the target domain that were not seen during training.

The new MLM stage is applied to a dataset containing all target domain data available, and an equal amount of source domain data (if the source domain dataset is smaller than the target domain dataset, all source domain data is used). 10 instances of each sample in this dataset are created, each with 15% of the tokens replaced with `[MASK]` tokens, following the same procedure as in the original BERT paper [21]. The model is then trained to predict each masked word.

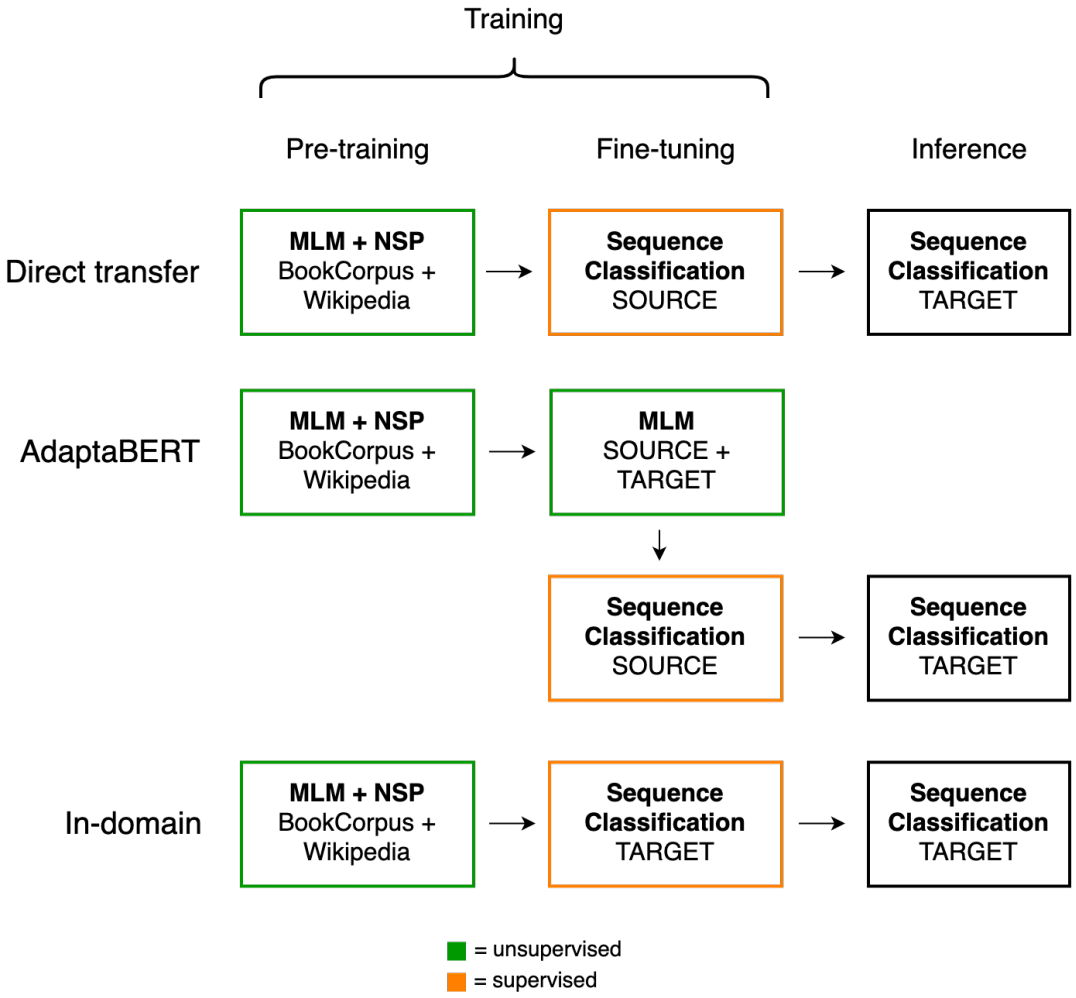


Figure 5.3: The stages of AdaptaBERT compared to direct transfer and in-domain BERT approaches

A diagram comparing the AdaptaBERT approach to direct transfer and in-domain approaches is shown in Figure 5.3. We can see AdaptaBERT only adds an extra unsupervised stage, so no extra labelled data is needed in either the source or target domain.

In Section 4.3 we found the Jaccard distance between articles and comments in our dataset was 0.44, indicating good scope for domain adaptation due to a shared vocabulary. However, each of these domains also has a large amount of OOV terms (130,123 for articles and 80,594 for comments). Therefore, our hypothesis is that AdaptaBERT will improve classification performance dramatically over direct transfer.

5.3.1 Experimental setup and evaluation

We modify the code provided by the authors to run AdaptaBERT on our cross-domain Reddit dataset [63]. The new MLM stage is very memory-intensive - we are limited to maximum sequence lengths of 128 tokens instead of the standard 512, as beyond this requires more memory than our hardware can supply. We use a learning rate of $5 \cdot 10^{-5}$ and a linear learning rate warm-up period

of 10% of training time as in the AdaptaBERT paper. However we find training with a batch size of 32 for 5 epochs for both the MLM and sequence classification fine-tuning tasks yields better results than the batch size of 64 for 3 epochs as used in the original paper.

Similarly to the previous section, we report results for *articles* → *comments* transfer as well as *comments* → *articles* transfer. We compare these against the direct transfer and in-domain baselines for the two domains from Section 5.2. Results are shown in Table 5.2 to 3.s.f.

Articles → comments transfer		
Model Type	**Macro-F1**	**Accuracy**
Direct transfer	54.4	51.8
AdaptaBERT	50.1	55.1
In-domain	67.2	67.6

Comments → articles transfer		
Model Type	**Macro-F1**	**Accuracy**
Direct transfer	69.6	69.3
AdaptaBERT	69.7	69.8
In-domain	76.2	76.5

Table 5.2: Evaluation metrics comparing AdaptaBERT to direct transfer and in-domain baselines

We can see AdaptaBERT gives slightly mixed results for *articles* → *comments* transfer, outperforming in terms of accuracy by around 4%, however letting F1 suffer by a similar amount. In the *comments* → *articles* case AdaptaBERT outperforms direct transfer slightly, but not in a statistically significant way. We can see in both directions, both direct transfer and AdaptaBERT fall short of the in-domain upper bound, however again the difference is much less in the *comments* → *articles* case.

5.3.2 Finding the optimal proportion of source domain examples for MLM

In standard AdaptaBERT, an equal amount of source domain and target domain examples are used for the novel MLM stage. We experiment with varying the amount of source domain content used relative to target domain content to see if we can improve classifier accuracy.

We evaluate AdaptaBERT for both *articles* → *comments* transfer and *comments* → *articles* transfer, with source proportion values of 0, $\frac{1}{3}$, $\frac{2}{3}$ and 1 (where 'source proportion' = number of source domain examples as a proportion of target domain examples). For each experiment we always use all target domain data available. Results are shown in Figure 5.4.

We can see for both directions of transfer, using $\frac{1}{3}$ of the number of target domain examples for the source domain provides the both the best F1 score and accuracy. Compared to using an equal amount of source and target domain examples (a source proportion of 1), this boosts F1 scores by around 8% to 58.0% for *articles* → *comments* transfer and by around 4% to 74.0% for *comments* → *articles* transfer. This is a significant improvement over the results in Table 5.2. Note that in the *comments* → *articles* case, this puts us only around 2% short of the in-domain baseline. Our new best AdaptaBERT results are shown in Table 5.3.

There is a general trend of increasing deterioration in classification performance the more source domain material is added to the MLM stage - this shows us classification performance is highest when the amount of source domain material used is fairly low compared to target domain material. This makes sense, since after a point adding more source domain material could attune BERT's internal contextualised embeddings too much towards source domain material, and could cause it to start 'forgetting' some target domain context. This is linked to a wider problem in neural-network-based models called 'catastrophic forgetting' [64].

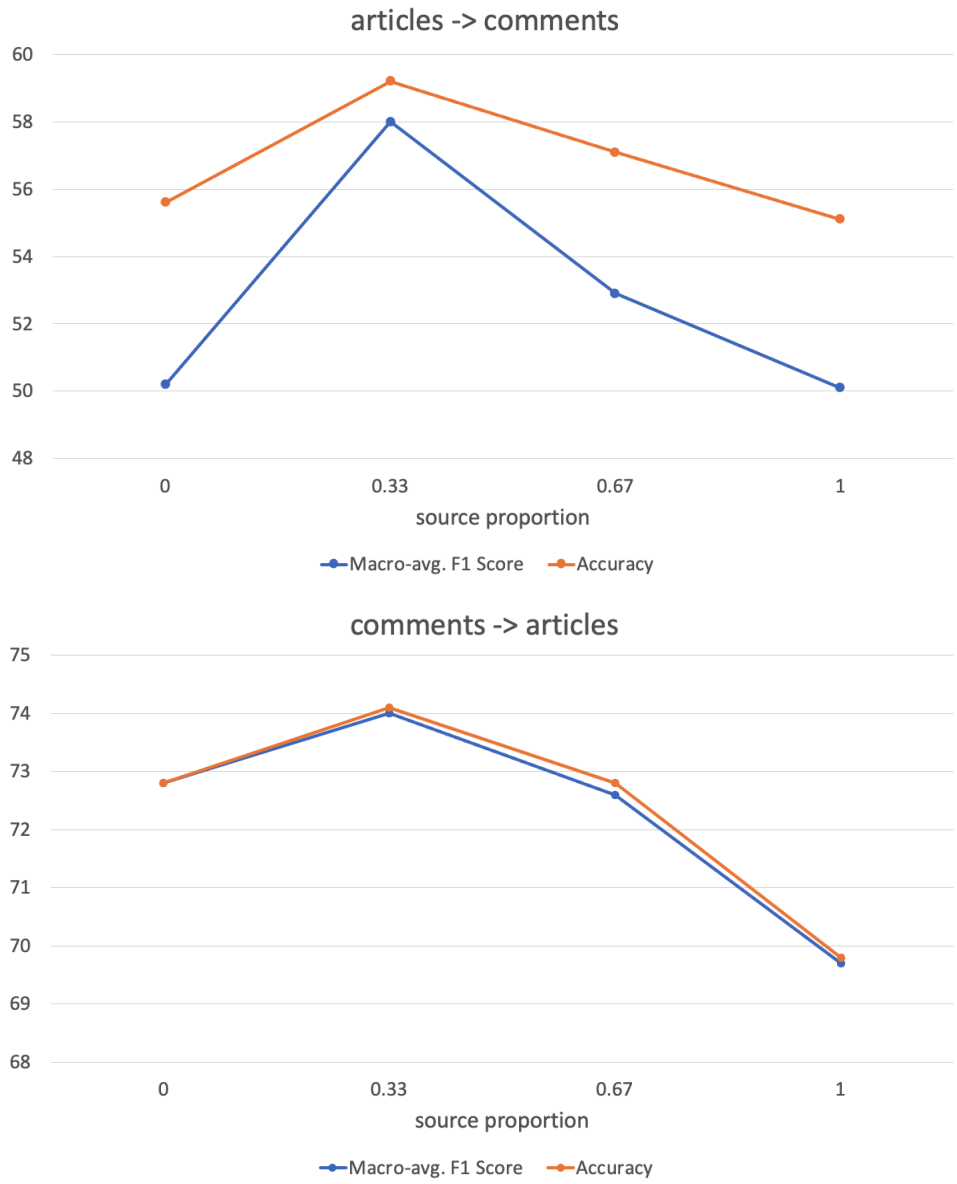


Figure 5.4: Performance of AdaptaBERT when varying proportion of source domain examples compared to target domain examples used in the MLM stage

5.4 Discussion

In this chapter, we have seen that direct transfer methods don't outperform in-domain (standard) methods for political bias classification in news and in social media. We show that AdaptaBERT improves over direct transfer when optimising for the right source:target domain ratio in AdaptaBERT's MLM stage, although it still can't match in-domain methods. That being said, classification performance is much better for *comments* → *articles* transfer than vice versa, indicating social media comments can be a good predictor for bias in news content.

In the original AdaptaBERT paper [48], Han & Eisenstein implement AdaptaBERT for a named entity recognition task on social media, examining transfer from news text to Twitter content. They find AdaptaBERT gives a 1% improvement in F1 score over direct transfer (58.9% vs 57.7%), and this extends to 3% when AdaptaBERT undergoes MLM on 1 million extra tweets. In our case, further work could be done to collect larger amounts of Reddit data and experiment running AdaptaBERT again - our Reddit dataset is fairly small, with just over 10,000 comments.

In Chapter 6 we will explore extending AdaptaBERT with a Next Sentence Prediction stage, to

Articles → comments transfer		
Model Type	**Macro-F1**	**Accuracy**
Direct transfer	54.4	51.8
AdaptaBERT	**58.0**	**59.2**
In-domain	67.2	67.6

Comments → articles transfer		
Model Type	**Macro-F1**	**Accuracy**
Direct transfer	69.6	69.3
AdaptaBERT	**74.0**	**74.1**
In-domain	76.2	76.5

Table 5.3: Evaluation metrics comparing AdaptaBERT with optimal source:target ratio in the MLM stage to direct transfer and in-domain baselines. Metrics that have improved since optimisation are highlighted in **bold**.

see if classification performance can be improved further.

Chapter 6

Extending AdaptaBERT with Next Sentence Prediction

In Chapter 5 we described the AdaptaBERT model, and measured its performance for the political bias detection task on our cross-domain Reddit dataset. We now extend AdaptaBERT by adding a Next Sentence Prediction stage to its fine-tuning stages, and assess the performance of the new model on political bias detection and named entity recognition tasks.

6.1 Motivation

Currently, AdaptaBERT only adds an extra masked language modelling objective to the training stages of BERT. However, standard BERT undergoes two pretraining objectives to train its contextualised word embeddings - masked language modelling (MLM) and next sentence prediction (NSP). The original BERT paper [21] states that the MLM objective is used to train a deep bidirectional internal representation of the words, and NSP is used mainly to help improve BERT's performance on question answering (QA) and natural language inference (NLI) tasks.

Intuitively this makes sense - tuning BERT's internal representations to better understand context between sentences will help it perform better at tasks which require scanning long sequences of text, such as QA and NLI. However, our cross-domain dataset of articles and Reddit comments also contains samples with long text sequences - each article in our dataset contains 31 sentences on average. Comments are less sentence-heavy, with each comment containing on average 2 sentences. Both domains show heavy right-hand skew in the distribution of number of sentences (see Figure 6.1).

We explore extending AdaptaBERT by adding a Next Sentence Prediction stage directly after the MLM stage, to see if this improves classification performance. Our hypothesis is that this will improve performance in the *comments* → *articles* scenario, since in this case the target domain contains textual content with many sentences per sample (31 on average), however the performance benefit will not be so great for *articles* → *comments*, since the target domain in this case doesn't exhibit many sentences per sample. The architecture we propose is shown in Figure 6.2, compared against standard AdaptaBERT and previous approaches.

6.2 Implementing next sentence prediction

In the original BERT paper [21], NSP is implemented by creating sentence pair examples `<A, B>` out of the pre-training corpus where 50% of the time sentence B is actually the sentence that follows A in the corpus, and the other 50% of the time it is a randomly-selected sentence. BERT is then trained for a binary classification objective to determine for each sentence pair whether sentence B actually follows sentence A or not. We mirror this methodology for our extended AdaptaBERT.

Similarly to standard AdaptaBERT, we use an equal amount of source domain and target domain examples in the NSP stage, up to a maximum of the entire target domain. For each article and

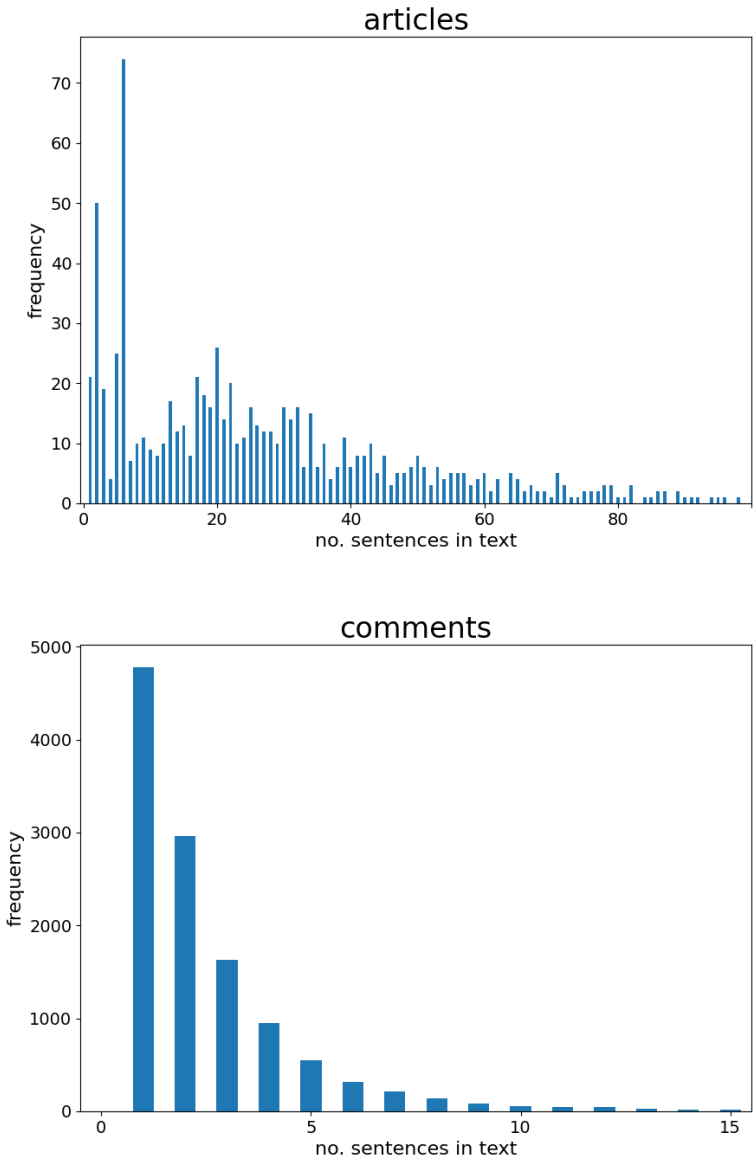


Figure 6.1: Distribution of number of sentences in each article (top) and comment (bottom) in our Reddit dataset

comment in our dataset, we choose 2 sentence pairs where the second sentence actually follows the first sentence, and 2 pairs where the second sentence is instead a random sentence from the same text. This gives us a collection of NSP training examples 4x larger than the size of the target domain.

6.3 Experimental setup

We compare the performance of extended AdaptaBERT to standard AdaptaBERT in two scenarios - political bias detection with the Reddit dataset as we have been doing previously, and also the named entity recognition (NER) task explored by Han & Eisenstein in the original AdaptaBERT paper [48].

The NER task is an example of a token classification task - so far, we have only explored sequence classification for political bias detection. Han & Eisenstein explore transfer from news content to tweets, using the CoNLL-2003 and WNUT 2016 NER datasets. The CoNLL dataset [65] is a corpus of 1,393 Reuters news stories [66] taken between August 1996 and August 1997, and the

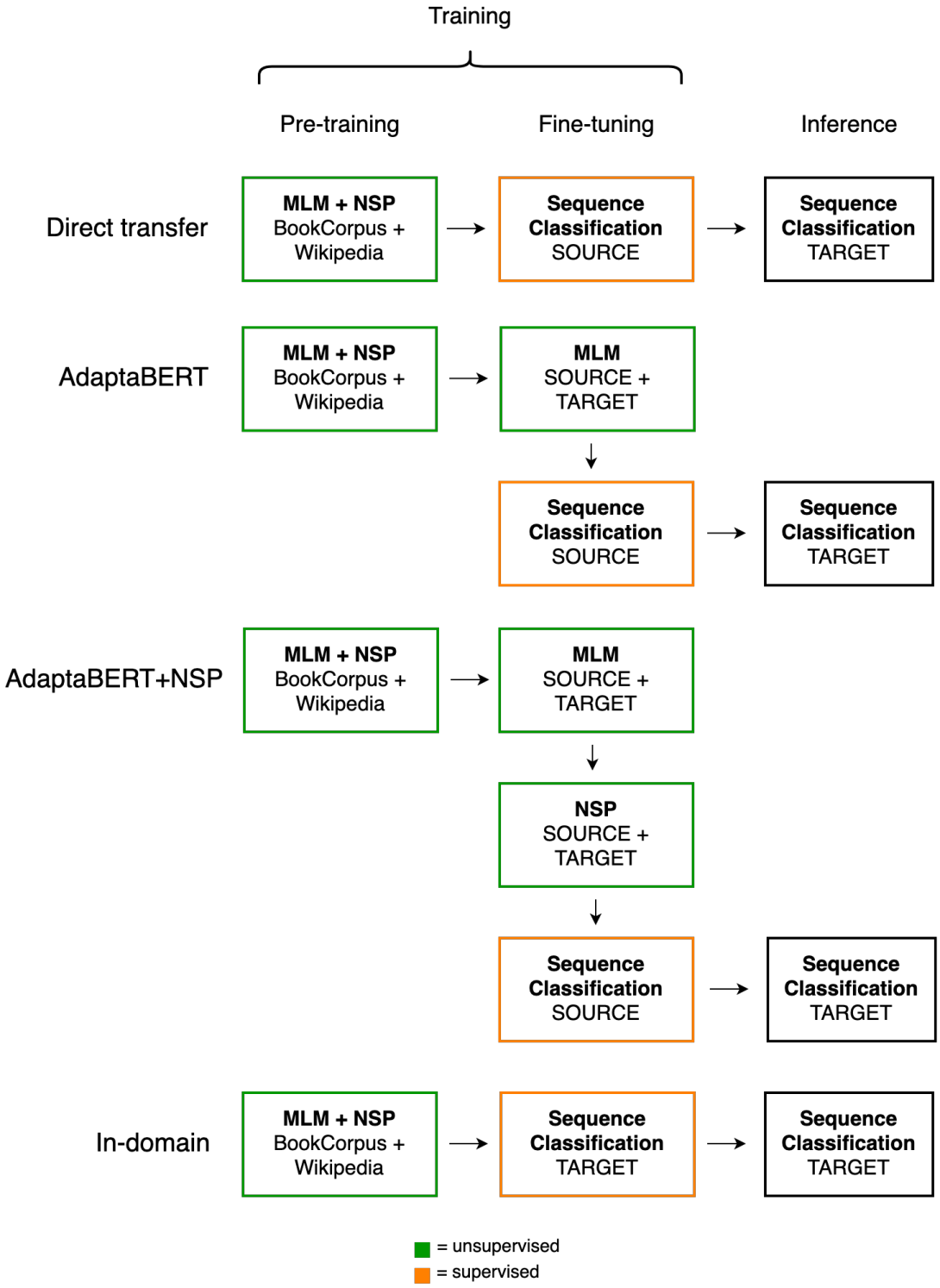


Figure 6.2: AdaptaBERT extended with the NSP stage, compared to standard AdaptaBERT, direct transfer and in-domain approaches

WNUT NER dataset [67] (taken from the Workshop of Noisy User Text 2016) is a collection of 3,819 generic tweets sampled from between December 2014 and February 2015. Both datasets are annotated at the token level by named entity (e.g. company, person, TV show, etc).

When tuning hyperparameters, we achieve best performance on both tasks when using only 1 epoch for the NSP stage. We use 5 epochs for MLM and sequence classification fine-tuning, as in Section 5.3. We also use a source proportion of $\frac{1}{3}$ for the MLM stage, as we found in Section 5.3.2 that this gives the best performance for standard AdaptaBERT.

6.4 Evaluation for political bias detection task

As in Chapter 5, we report results for both *articles* → *comments* transfer and *comments* → *articles* transfer. Results are shown in Table 6.1.

Task	Bias detection	
Transfer direction	Articles → comments	
Model Type	**Macro-F1**	**Accuracy**
Direct transfer	54.4	51.8
AdaptaBERT	58.0	59.2
AdaptaBERT+NSP	58.0	58.1
In-domain	67.2	67.6

Task	Bias detection	
Transfer direction	Comments → articles	
Model Type	**Macro-F1**	**Accuracy**
Direct transfer	69.6	69.3
AdaptaBERT	74.0	74.1
AdaptaBERT+NSP	70.9	71.0
In-domain	76.2	76.5

Table 6.1: Evaluation metrics comparing extended AdaptaBERT to standard AdaptaBERT, direct transfer and in-domain baselines for the political bias detection task

We can see that in both transfer directions, the extended AdaptaBERT fails to improve over the standard AdaptaBERT in both F1 score and accuracy, although it is able to match F1 score in the *articles* → *comments* case. In the *comments* → *articles* case, performance almost deteriorates to the same level as direct transfer. This suggests our hypothesis that adding Next Sentence Prediction would help classifier performance is false, regardless of which domain is the target domain. Our results seem to indicate that performance deteriorates the most when the domain with most sentences per example (i.e. articles) is the target domain, however more experiments on different datasets are ideally needed to test this hypothesis.

6.5 Evaluation for NER task

We only evaluate *articles* → *tweets* transfer, since this is the only transfer direction evaluated by Han & Eisenstein [48]. They also only provide F1 score and not accuracy. Results are shown in Table 6.2.

We can see the extended AdaptaBERT improves over AdaptaBERT by 2.5% in F1 score, and comes within 3% of the in-domain baseline.

Examining the distribution of sentences in the NER dataset (see Figure 6.3), we can see in fact both domains exhibit a low number of sentences per example. The articles, rather than having been preserved as single examples, have been split up into roughly 1 sentence per example by the WNUT dataset authors. A minority of examples contain 2 or more sentences. The average number

Task	NER
Transfer direction	Articles → tweets
Model Type	**Macro-F1**
Direct transfer	57.7
AdaptaBERT	58.9
AdaptaBERT+NSP	61.4
In-domain	64.3

Table 6.2: Evaluation metrics comparing extended AdaptaBERT to standard AdaptaBERT, direct transfer and in-domain baselines for the NER task. Results 1, 2, and 4 taken from [48].

of sentences in each tweet is of course quite low, due to Twitter's 140-character tweet limit as of December 2016, the month the dataset was published.

This shows us that AdaptaBERT extended with NSP can still improve over standard AdaptaBERT even when both the source and target domain exhibit a low number of sentences per training example. Based on these results and those in Section 6.4, we hypothesis that NSP improves classifier performance when the source and target domain have **similar** distributions of number of sentences per example, irrespective of how high or low the average number of sentences per example actually is. More experimentation on different datasets is needed to fully test this hypothesis.

6.6 Finding the optimal proportion of source domain examples for NSP

In Sections 6.4 and 6.5, we used an equal number of source and target domain examples for the Next Sentence Prediction stage. We now aim to find which ratio of source to target domain examples gives the optimal performance of extended AdaptaBERT, similarly to our experiments with the MLM stage in standard AdaptaBERT in Section 5.3.2. Again, we trial source proportion values of 0, $\frac{1}{3}$, $\frac{2}{3}$ and 1. In all cases we use all target domain data available.

Results for the political bias task are shown in Figure 6.4. We can see in the *articles* → *comments* case we achieve the best metrics with a source proportion of 1 (i.e. using an equal amount of source and target domain data). Interestingly, in the *comments* → *articles* case we achieve the best metrics using no source domain material at all in the NSP stage (the F1 line is hidden behind the accuracy line between source proportions of 0 and $\frac{1}{3}$). This gives a final F1 score of 73.5%, compared to 70.9% when using an equal proportion. This almost matches the performance of standard AdaptaBERT, however does not exceed it. Our results don't appear to follow the same pattern seen in Section 5.3.2, where adding more source domain material leads to increasing deterioration in model performance. In fact, in both transfer directions using an equal amount of source and target domain material in the NSP stage results in very good classifier performance compared to other ratios. Our new best extended AdaptaBERT results for this task are shown in Table 6.3.

Results for the NER task are shown in Figure 6.5. We only evaluate *articles* → *tweets* in this case. We can see we achieve best performance with a source proportion of $\frac{1}{3}$, similarly to our experiments in Section 5.3.2. This gives a final F1 score of 62.2%, which is a 0.8% improvement over the result in Section 6.5. Overall, this is a 3.3% improvement over standard AdaptaBERT for this task. Our new best extended AdaptaBERT results for this task are shown in Table 6.4.

6.7 Discussion

In this chapter we have shown that extending AdaptaBERT with a Next Sentence Prediction stage yields mixed results - we achieve an improvement in classification accuracy for an NER task, but performance deterioration for the original political bias detection task.

There are a number of ways we could extend our research here. We could experiment with having

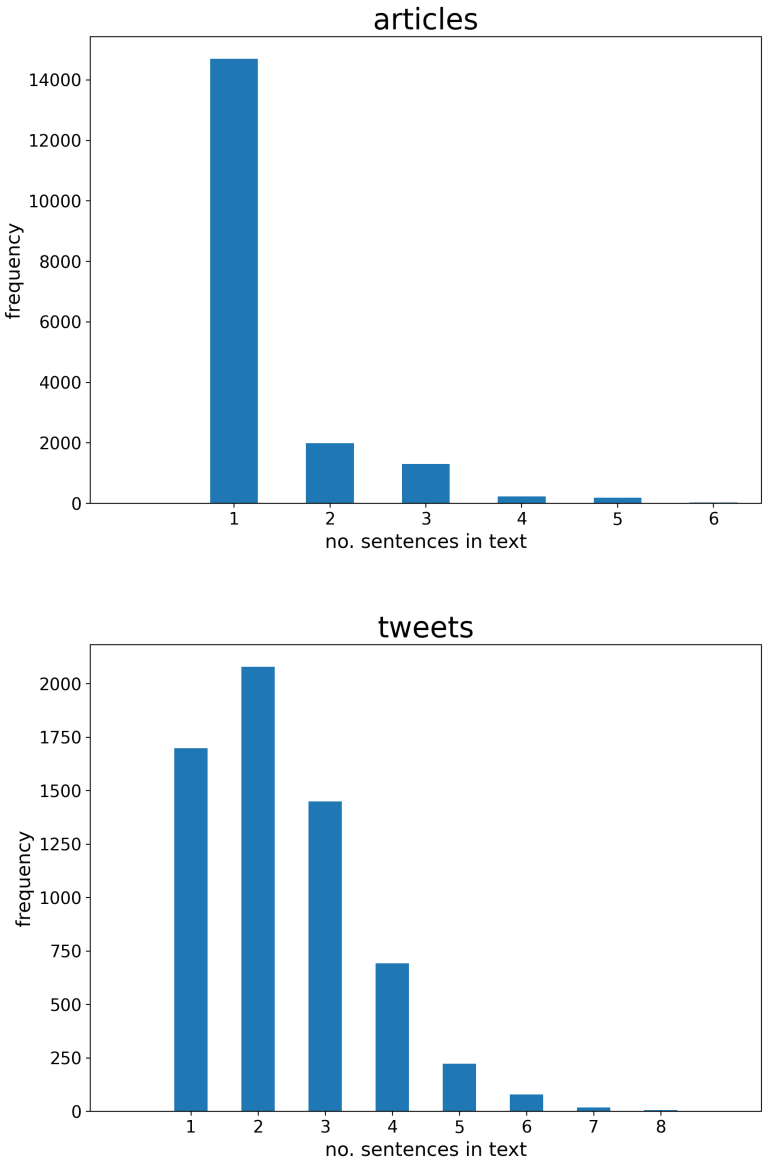


Figure 6.3: Distribution of number of sentences in each article (top) and tweet (bottom) in the WNUT 2016 NER dataset

more source domain than target domain material in the NSP stage - we only evaluate using no source domain material through to matching the amount of target domain material present. We could also evaluate our extended AdaptaBERT model on the POS tagging task used in the original paper [48]. As mentioned in Section 5.3.1, Han & Eisenstein extend their MLM training corpus with an extra 1 million unlabeled tweets, which provides an extra boost in performance. Further work can be done evaluating our extended model with a larger scale cross-domain Reddit dataset. We could also experiment further with our hypothesis in Section 6.5 that NSP can improve performance when the source and target domain have a similar number of sentences per training example.

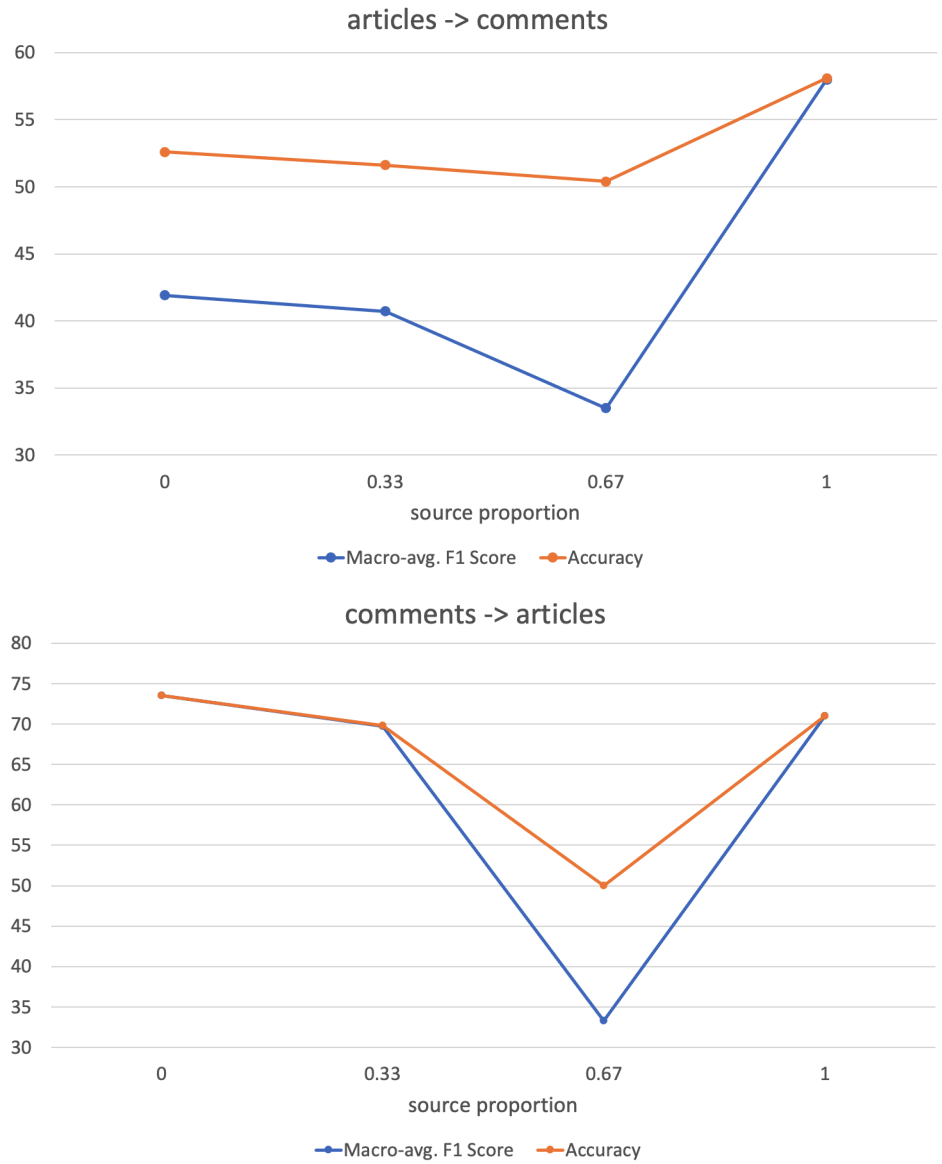


Figure 6.4: Performance of extended AdaptaBERT on the political bias detection task when varying proportion of source domain examples compared to target domain examples used in the NSP stage

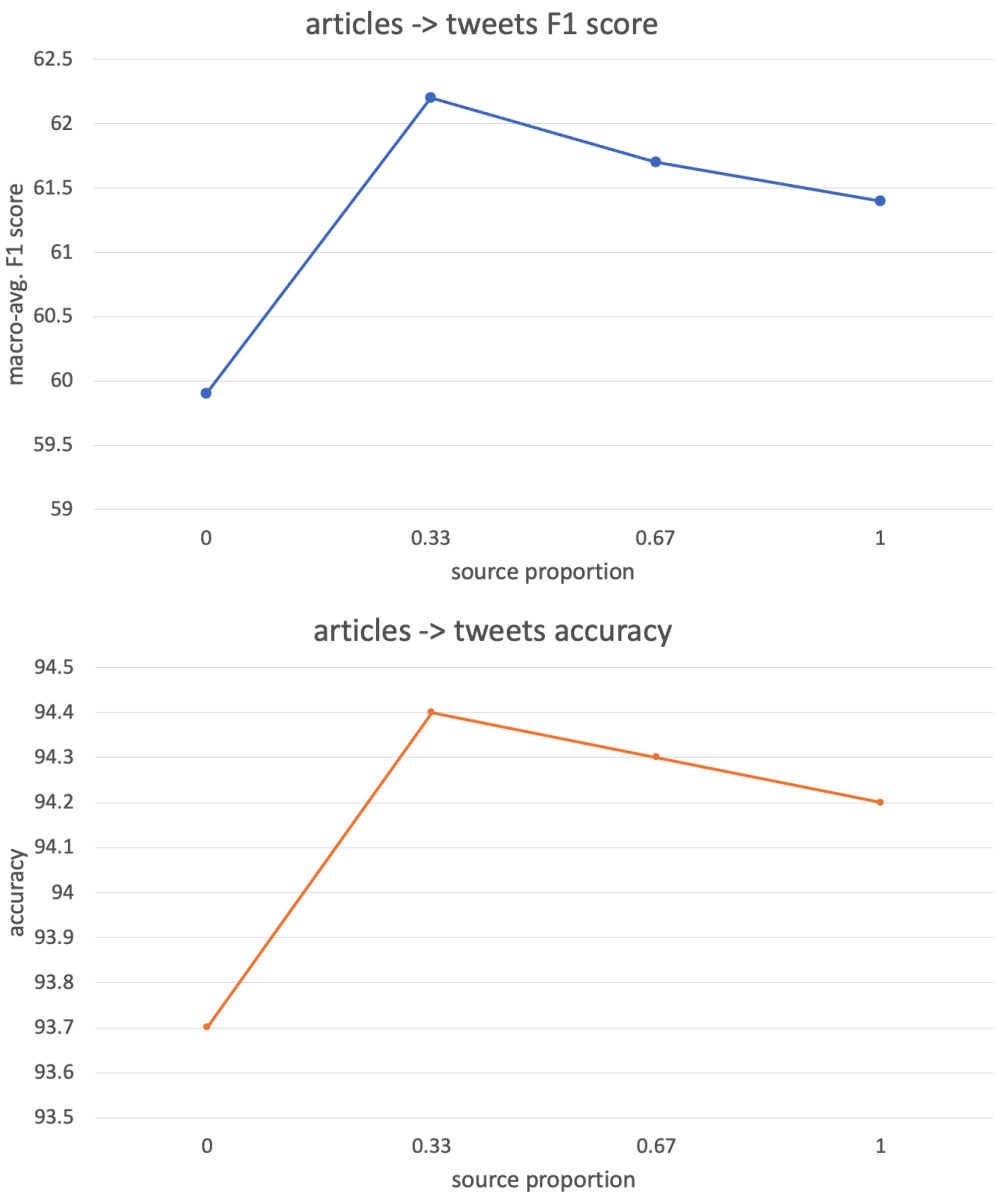


Figure 6.5: Performance of extended AdaptaBERT on the NER task when varying proportion of source domain examples compared to target domain examples used in the NSP stage

Task	Bias detection	
Transfer direction	Articles → comments	
Model Type	**Macro-F1**	**Accuracy**
Direct transfer	54.4	51.8
AdaptaBERT	58.0	59.2
AdaptaBERT+NSP	58.0	58.1
In-domain	67.2	67.6

Task	Bias detection	
Transfer direction	Comments → articles	
Model Type	**Macro-F1**	**Accuracy**
Direct transfer	69.6	69.3
AdaptaBERT	74.0	74.1
AdaptaBERT+NSP	**73.5**	**73.5**
In-domain	76.2	76.5

Table 6.3: Evaluation metrics comparing extended AdaptaBERT with optimal source:target ratio in the NSP stage to standard AdaptaBERT, direct transfer and in-domain baselines for the political bias detection task. Metrics that have improved since optimisation are highlighted in **bold**.

Task	NER
Transfer direction	Articles → tweets
Model Type	**Macro-F1**
Direct transfer	57.7
AdaptaBERT	58.9
AdaptaBERT+NSP	**62.2**
In-domain	64.3

Table 6.4: Evaluation metrics comparing extended AdaptaBERT with the optimal source:target ratio in the NSP stage to standard AdaptaBERT, direct transfer and in-domain baselines for the NER task. Results 1, 2, and 4 taken from [48]. Metrics that have improved since optimisation are highlighted in **bold**.

Chapter 7

Conclusion & Future Work

7.1 Conclusion

In this project we investigated whether transfer learning, specifically domain adaptation, can help detect political bias on social media, using knowledge learnt from detecting the same bias in news content. We started by surveying classifiers to detect bias in the news, finding SVMs and gradient-boosted forests to give the best performance, using features from earlier research. We then examined BERT models for the same task, which perform feature extraction and classification holistically, finding these also give fairly good performance.

Using our self-developed cross-domain dataset of news articles and related Reddit comments, we explored unsupervised domain adaptation techniques for detecting political bias in news and social media comments. We found classification performance when using social media comments as a predictor for news bias (*comments* → *articles* transfer) significantly outperforms the reverse scenario, however in both cases transfer learning cannot outperform standard in-domain classifiers. We also found that a state-of-the-art domain adaptive BERT model called AdaptaBERT outperforms direct transfer learning when optimising for the right mix of source and target domain material during training, coming very close to matching the performance of in-domain classifiers.

We investigated extending AdaptaBERT with an extra Next Sentence Prediction stage, originally used in BERT pre-training, to see if this improves classification performance. This extension yields a 3.3% increase in F1 score for an NER task originally explored in the AdaptaBERT paper [48], however this extension also results in performance deterioration for the original political bias detection task, possibly due to the differences between the source and target domains in terms of number of sentences per sample.

Further work still needs to be done in the field of unsupervised domain adaptation to fully match the performance of in-domain classifiers, and to see if in-domain performance can be exceeded. This would aid political bias detection on social media tremendously, where standard supervised classification cannot be performed due to the lack of sufficient annotated data.

7.2 Ethical Issues

With regards to data protection, our project does not involve collecting any personal data from Reddit users - all information is collected via the public Reddit API, and we immediately discard comment author information from our datasets before running our models, since we don't make use of this information in any way. An earlier project idea involved analysing the distribution of left-wing and right-wing users across various subreddits - this would involve inferring political affiliations on a per-user basis, which counts as processing 'special-category' data under the UK Data Protection Act (DPA) 2018 [68]. In this case we would need to take extra precautions when storing and processing this sensitive information.

One issue is that a third party could take our bias detectors and apply them on comments all from one social media user to determine that user's political leaning, which would of course be inferring

sensitive data. This kind of privacy attack could be performed with any bias-detection machine learning model. If our model were deployed in a production environment, we could limit this kind of attack by placing our model behind a query-based interface and limiting the number of queries that can be placed.

In using Reddit comments as a predictor for political bias in the wider news context, our model may unknowingly be biased towards writing styles used by the demographics in our dataset. Reddit is known to have a mainly young, American, white and male userbase [69], and so a bias detection classifier trained on Reddit text such as ours may start to mispredict if it comes across news or comments with writing styles different to those used by this demographic online.

This risk is increased when using large language models such as BERT, which are pre-trained on huge amounts of data - very little is understood about the exact mechanisms BERT uses during training to ensure classification success [70]. Jo & Gebru [71] carried out a case study on GPT-2, a large language model similar to BERT, and its pre-training corpus of Reddit data. They noted Reddit's demographics are not representative of demographics in the wider world, and due consideration of this had not been taken by the GPT-2 creators before publishing their model. We have attempted to publish significant information about our Reddit dataset in Chapter 4, including which subreddits we sourced information from and how much data from each subreddit, to make our training data collection methodology more transparent.

7.3 Future Work

In this section we propose several ideas for future work that could take our research further.

- **Collecting more training data**: Our cross-domain Reddit dataset is fairly small, with 806 news articles and 11,954 Reddit comments. Our research would be made much more reliable with a larger collection of news articles in our dataset, plus a wider range of politically-themed subreddits to source comments from. Many of the posts we scraped from Reddit were actually links to screenshots of news articles and tweets rather than actual articles, which we discarded - keeping these screenshots and extracting text from them would also give us access to much more Reddit training data.
- **Using more modern classifiers**: We chose BERT models for our experiments in Chapters 5 and 6 since BERT performs feature extraction automatically, and still provides good classification performance for detecting political bias as seen in Chapter 3. Newer language models, such as XLNet [72], claim to improve upon BERT's performance significantly, and so assessing their performance on this political bias detection task may yield superior results.
- **Exploring a multi-modal approach**: Throughout this project we only examine textual content in order to detect bias. Given that much of the data we scraped from Reddit is in image format, we could build a multi-modal model that fuses information learnt from images (e.g. with a convolutional neural network) and information learnt from the text to detect bias.
- **Assessing non-political subreddits**: In this project we only considered politically-themed subreddits to train our classifier. Kane & Luo [36] explored whether non-political subreddits still exhibit some unconscious political leaning, using topic modelling with Latent Dirichlet Allocation. Using our classifiers trained on political subreddit data, we could examine the same task but with a transfer learning approach, to see if our results differ from Kane & Luo's.
- **Exploring bias across time and geography**: Our data is mainly focused around US news sources and comments from Reddit users in the USA, due to Reddit's overwhelmingly American userbase. Further work can be done exploring how bias scales vary between countries, e.g. by looking at social platforms that span a more global audience such as WhatsApp [73], and also how bias has changed throughout history.
- **A browser extension for assessing echo chambers**: An online echo chamber is an environment where a person only encounters information that reflects and reinforces their own opinions [74]. Social media users nowadays are more aware of how echo chambers can affect their point of view. By developing a browser extension that can analyse the text on

screen for political bias using our classifiers, social media users can examine how the content they see on social media is politically oriented, to see if they are inside an echo chamber.

- **Tackling other tasks with Next Sentence Prediction**: In this project we only examined the effect of adding Next Sentence Prediction to AdaptaBERT for political bias detection and named entity recognition tasks, both of which are classification tasks. Further work could be done exploring how Next Sentence Prediction could be utilised for domain-adaptive versions of other tasks such as question answering and natural language inference.

Bibliography

[1] Pew Research Center. *U.S. Media Polarization and the 2020 Election: A Nation Divided*. Available from: https://www.journalism.org/2020/01/24/u-s-media-polarization-and-the-2020-election-a-nation-divided/ [Accessed 07/06/2021]

[2] Pew Research Center. *Publics Globally Want Unbiased News Coverage, but Are Divided on Whether Their News Media Deliver*. Available from: https://www.pewresearch.org/global/2018/01/11/publics-globally-want-unbiased-news-coverage-but-are-divided-on-whether-their-news-media-deliver/ [Accessed 11/01/2021]

[3] Pew Research Center. *News Use Across Social Media Platforms 2018*. Available from: https://www.journalism.org/2018/09/10/news-use-across-social-media-platforms-2018/ [Accessed 24/05/2021]

[4] Media Bias/Fact Check. *Media Bias/Fact Check*. Available from: https://mediabiasfactcheck.com/ [Accessed 10/01/2021]

[5] Ad Fontes Media. *Interactive Media Bias Chart*. Available from: https://www.adfontesmedia.com/interactive-media-bias-chart-2/ [Accessed 10/01/2021]

[6] Ad Fontes Media. *Methodology Overview*. Available from: https://www.adfontesmedia.com/how-ad-fontes-ranks-news-sources/ [Accessed 10/06/2021]

[7] W. Wang. *Calculating Political Bias and Fighting Partisanship with AI*. Available from: https://www.thebipartisanpress.com/politics/calculating-political-bias-and-fighting-partisanship-with-ai/ [Accessed 17/02/2021]

[8] R. Baly, G. Karadzhov, D. Alexandrov, J. Glass, P. Nakov. Predicting Factuality of Reporting and Bias of News Media Sources. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2018. Available from: https://www.aclweb.org/anthology/D18-1389/ [Accessed 14/01/2021]

[9] M. Smith. *Left-wing vs right-wing: it's complicated*. Available from: https://yougov.co.uk/topics/politics/articles-reports/2019/08/13/left-wing-vs-right-wing-its-complicated [Accessed 11/02/2021]

[10] Diffen. *Left Wing vs Right Wing*. Available from: https://www.diffen.com/difference/Left_Wing_vs_Right_Wing [Accessed 11/02/2021]

[11] Media Bias/Fact Check. *Methodology*. Available from: https://mediabiasfactcheck.com/methodology/ [Accessed 14/01/2021]

[12] Media Bias Fact Check. *CNN*. Available from: https://mediabiasfactcheck.com/cnn/ [Accessed 14/01/2021]

[13] R. Baly, G. Karadzhov, D. Alexandrov, J. Glass, P. Nakov. *Factuality and Bias Prediction of News Media*. Available from: https://github.com/ramybaly/News-Media-Reliability [Accessed 14/01/2021]

[14] T. Mikolov, K. Chen, G. Corrado, J. Dean. Efficient Estimation of Word Representations in Vector Space. *Proceedings of the International Conference on Learning Representations (ICLR)*, January 2013. Available from: https://arxiv.org/pdf/1301.3781.pdf [Accessed 15/01/2021]

[15] J. Pennington, R. Socher, C. Manning. GloVe: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, October 2014. pp. 1532-1543. Available from: https://nlp.stanford.edu/pubs/glove.pdf [Accessed 15/01/2021]

[16] T. Bolukbasi, K. Chang, J. Zou, V. Saligrama, A. Kalai. Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. *30th Conference on Neural Information Processing Systems*, 2016. Available from: https://arxiv.org/pdf/1607.06520v1.pdf [Accessed 16/01/2021]

[17] J. Gordon, M. Babaeianjelodar, J. Matthews. Studying Political Bias via Word Embeddings. *Companion Proceedings of the Web Conference*, 2020. Available from: https://dl.acm.org/doi/pdf/10.1145/3366424.3383560 [Accessed 16/01/2021]

[18] Y. Freund, R. Schapire. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 1997. Vol. 55, pp. 119-139. Available from: https://www.sciencedirect.com/science/article/pii/S002200009791504X [Accessed 13/06/2021]

[19] J. Friedman. Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics*, October 2001. Vol. 29, No. 5, pp. 1189-1232. Available from: https://www.jstor.org/stable/2699986 [Accessed 13/06/2021]

[20] M. Voong, K. Gunda, SS. Gokhale. Predicting the Political Polarity of Tweets Using Supervised Machine Learning. *IEEE 44th Annual Computers, Software, and Applications Conference*, 2020 Jul 13. p1707-1712. Available from: https://ieeexplore.ieee.org/document/9202835 [Accessed 20/01/2021]

[21] J. Devlin, M. Chang, K. Lee, K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, June 2019. Vol. 1, pp. 4171-4186. Available from: https://arxiv.org/pdf/1810.04805v2.pdf [Accessed 17/01/2021]

[22] S. Chun, R. Holowczak, K. Dharan, R. Wang, S. Basu, J. Geller. Detecting Political Bias Trolls in Twitter Data. *Proceedings of the 15th International Conference on Web Information Systems and Technologies*, 2019. Available from: https://www.scitepress.org/Link.aspx?doi=10.5220%2f0008350303340342 [Accessed 17/01/2021]

[23] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. Available from: https://arxiv.org/pdf/1907.11692.pdf [Accessed 17/01/2021]

[24] R. Baly, G. Karadzhov, J. An, H. Kwak, Y. Dinkov, A. Ali, J. Glass, P. Nakov. What Was Written vs. Who Read It: News Media Profiling Using Text Analysis and Social Media Context. *Association for Computational Linguistics*, 2020. Available from: https://www.aclweb.org/anthology/2020.acl-main.308/ [Accessed 19/01/2021]

[25] W. Sattelberg. *The Demographics Of Reddit: Who Uses The Site?*. Available from: https://social.techjunkie.com/demographics-reddit/ [Accessed 21/01/2021]

[26] NORDLAN. *The evidence is piling up: Trump's iron grip over the GOP has been institutionalized*. Available from: https://www.reddit.com/r/Liberal/comments/n6xqzm/the_evidence_is_piling_up_trumps_iron_grip_over/ [Accessed 14/06/2021]

[27] Reddit. *Reddit API: Documentation*. Available from: https://www.reddit.com/dev/api [Accessed 07/05/21]

[28] J. Baumgartner, S. Zannettou, B. Keegan, M. Squire, J. Blackburn. The PushShift Reddit dataset. *Proceedings of the International AAAI Conference on Web and Social Media*, 6 May 2020 (Vol. 14, pp. 830-839). Available from: https://arxiv.org/pdf/2001.08435v1.pdf [Accessed 09/05/21]

[29] B. Boe. *PRAW: The Python Reddit API Wrapper*. Available from: https://praw.readthedocs.io/en/latest/ [Accessed 30/05/2021]

[30] K. Lin, C. Yang, H. Chen. What emotions do news articles trigger in their readers? *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2007. Available from: https://dl.acm.org/doi/10.1145/1277741.1277882 [Accessed 19/01/2021]

[31] P. Ekman, R. Sorenson, W. Friesen. Pan-Cultural Elements in Facial Displays of Emotion. *Science*, 1969. Volume 164, p. 86-88. Available from: https://science.sciencemag.org/content/164/3875/86/tab-pdf [Accessed 19/01/2021]

[32] C. Strapparava, R. Mihalcea. SemEval-2007 Task 14: Affective Text. *Proceedings of the Fourth International Workshop on Semantic Evaluations*, 2007. p70-74. Available from: https://www.aclweb.org/anthology/S07-1013/ [Accessed 19/01/2021]

[33] C. Tan. Sentiment Analysis From the Reader's Perspective [MEng Thesis]. Imperial College London. 2020.

[34] V. Garimella, I. Weber. A Long-Term Analysis of Polarization on Twitter. *Proceedings of the International AAAI Conference on Web and Social Media*, 3rd May 2017. Vol. 11, No. 1. Available from: https://ojs.aaai.org/index.php/ICWSM/article/view/14918 [Accessed 20/01/2021]

[35] D. Rao, D. Yarowsky, A. Shreevats, M. Gupta. Classifying Latent User Attributes in Twitter. *Proceedings of the 2nd International Workshop on Search and Mining User-Generated Contents*, 2010. p37-44. Available from: https://dl.acm.org/doi/abs/10.1145/1871985.1871993 [Accessed 20/01/2021]

[36] B. Kane, J. Luo. Do the Communities We Choose Shape our Political Beliefs? A Study of the Politicization of Topics in Online Social Groups. *IEEE International Conference on Big Data*, 10th December 2018. p3665-3671. Available from: https://ieeexplore.ieee.org/document/8622535 [Accessed 20/01/2021]

[37] J. Massachs, C. Monti, G.D. Morales, F. Bonchi. Roots of Trumpism: Homophily and Social Feedback in Donald Trump Support on Reddit. *12th ACM Conference on Web Science*, 6th July 2020. pp. 49-58. Available from: https://dl.acm.org/doi/10.1145/3394231.3397894 [Accessed 25/05/2021]

[38] S. Ruder. *Neural Transfer Learning for Natural Language Processing* (dissertaion). 2019. Available from: https://ruder.io/thesis/neural_transfer_learning_for_nlp.pdf [Accessed 19/05/2021]

[39] S.J. Pan, Q. Yang. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 16th Oct 2009. Available from: https://ieeexplore.ieee.org/document/5288526 [Accessed 19/05/2021]

[40] Clker. *Free Clip Art & Images*. Available from: http://www.clker.com/ [Accessed 01/06/2021]

[41] H. Daumé III, D. Marcu. Domain Adaptation for Statistical Classifiers. *Journal of Artificial Intelligence Research*, 21st June 2006. Vol. 26, pp. 101-126. Available from: https://arxiv.org/pdf/1109.6341.pdf [Accessed 19/05/2021]

[42] J. Blitzer, R. McDonald, F. Pereira. Domain Adaptation with Structural Correspondence Learning. *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, July 2006. pp. 120-128. Available from: http://john.blitzer.com/papers/emnlp06.pdf [Accessed 20/05/2021]

[43] S.J. Pan, X. Ni, J.T. Sun, Q. Yang, Z. Chen. Cross-domain Sentiment Classification via Spectral Feature Alignment. *Proceedings of the 19th International Conference on World Wide Web*, 26th April 2010. pp. 751-760. Available from: https://dl.acm.org/doi/abs/10.1145/1772690.1772767 [Accessed 20/05/2021]

[44] L. Michelbacher, Q. Han, H. Schütze. Unsupervised Feature Adaptation for Cross-Domain NLP with an Application to Compositionality Grading. *International Conference on Intelligent Text Processing and Computational Linguistics*, 24th March 2013. pp. 1-12. Springer, Berlin, Heidelberg. Available from: https://rdcu.be/ckZ5Z [Accessed 20/05/2021]

[45] B. Myagmar, J. Li, S. Kimura. Cross-Domain Sentiment Classification with Bidirectional Contextualized Transformer Language Models. *IEEE Access*, 8th November 2019. Vol. 7, pp. 163219-30. Available from: https://ieeexplore.ieee.org/abstract/document/8894409 [Accessed 22/05/2021]

[46] H. Ye, Q. Tan, R. He, J. Li, H.T. Ng, L. Bing. Feature Adaptation of Pre-Trained Language Models across Languages and Domains with Robust Self-Training. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, November 2020. pp. 7386-7399. Available from: https://www.aclweb.org/anthology/2020.emnlp-main.599.pdf [Accessed 23/05/2021]

[47] C. Du, H. Sun, J. Wang, Q. Qi, J. Liao. Adversarial and Domain-Aware BERT for Cross-Domain Sentiment Analysis. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, July 2020. pp. 4019-4028. Available from: https://www.aclweb.org/anthology/2020.acl-main.370/ [Accessed 23/05/2021]

[48] X. Han, J. Eisenstein. Unsupervised Domain Adaptation of Contextualized Embeddings for Sequence Labeling. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, November 2019. pp. 4238–4248. Available from: https://www.aclweb.org/anthology/D19-1433.pdf [Accessed 23/05/2021]

[49] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, N.A. Smith. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. *Proceedings of the 58th Annual Meeting of the Association forComputational Linguistics (ACL)*, July 2020. Available from: https://www.aclweb.org/anthology/2020.acl-main.740.pdf [Accessed 23/05/2021]

[50] F. Pedregosa, G. Varoquaux et al. *scikit-learn*. Available from: https://sklearn.org/ [Accessed 16/02/2021]

[51] J. Holwech, F. Shah. *NewsScraper*. Available from: https://github.com/fawazshah/NewsScraper [Accessed 29/03/2021]

[52] L. Ou-Yang. *Newspaper3k: Article scraping & curation*. Available from: https://newspaper.readthedocs.io/en/latest/ [Accessed 29/03/2021]

[53] S. Bird, E. Loper, E. Klein. Natural Language Processing with Python. O'Reilly Media Inc. 2009.

[54] Princeton University. *About WordNet*. Available from: https://wordnet.princeton.edu/. [Accessed 29/03/2021]

[55] I. Loshchilov, F. Hutter. Decoupled Weight Decay Regularization. 2017. Available from: https://arxiv.org/pdf/1711.05101.pdf [Accessed 29/03/2021]

[56] FiveThirtyEight. *Russian Troll Tweets*. Available from: https://github.com/fivethirtyeight/russian-troll-tweets/ [Accessed 06/04/21]

[57] Y. Wu, M. Schuster et al. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. 2016. Available from: https://arxiv.org/pdf/1609.08144.pdf [Accessed 08/04/21]

[58] R. Battle. *Weaknesses of WordPiece Tokenization*. Available from: https://medium.com/@rickbattle/weaknesses-of-wordpiece-tokenization-eb20e37fec99 [Accessed 25/05/2021]

[59] A. Guimaraes, O. Balalau, E. Terolli, G. Weikum. Analyzing the traits and anomalies of political discussions on Reddit. *Proceedings of the International AAAI Conference on Web and Social Media*, 6th July 2019. Vol. 13, p. 205-213. Available from: https://ojs.aaai.org/index.php/ICWSM/article/view/3222 [Accessed 10/05/21]

[60] C. Tyler. *Reddit's leftward political bias*. Available from: https://datainsights.pub/reddit_political_leanings/ [Accessed 14/05/21]

[61] J. Brownlee. *A Gentle Introduction to Concept Drift in Machine Learning*. Available from: https://machinelearningmastery.com/gentle-introduction-concept-drift-machine-learning/ [Accessed 17/05/21]

[62] D. Gaffney, J.N. Matias. Caveat Emptor, Computational Social Science: Large-Scale Missing Sata in a Widely-Published Reddit Corpus. *PloS One*, 6th July 2018. 13(7):e0200162. Available from: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0200162 [Accessed 21/05/2021]

[63] X. Han, J. Eisenstein, F. Shah. *Unsupervised Domain Adaptation of Contextualized Embeddings for Sequence Labeling*. Available from: https://github.com/fawazshah/AdaptaBERT/ [Accessed 28/05/2021]

[64] M. McCloskey, N.J. Cohen. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. *Psychology of Learning and Motivation*, 1st January 1989. Vol. 24, pp. 109-165. Academic Press. Available from: https://www.sciencedirect.com/science/article/pii/S0079742108605368 [Accessed 06/06/2021]

[65] E.F. Sang, F. De Meulder. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. *Proceedings of CoNLL-2003*, 12th June 2003, Edmonton, Canada. pp. 142-147. Available from: https://arxiv.org/pdf/cs/0306050v1.pdf [Accessed 04/06/2021]

[66] D. D. Lewis, Y. Yang, T. Rose, F. Li. *RCV1: A New Benchmark Collection for Text Categorization Research. Journal of Machine Learning Research*, 5:361-397, 2004. Available from: http://www.jmlr.org/papers/volume5/lewis04a/lewis04a.pdf [Accessed 04/06/2021]

[67] B. Strauss, B. Toma, A. Ritter, M.C. De Marneffe, W. Xu. Results of the WNUT16 Named Entity Recognition Shared Task. *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, December 2016. pp. 138-144. Available from: https://www.aclweb.org/anthology/W16-3919.pdf [Accessed 04/06/2021]

[68] Data Protection Act 2018 (UK). Available from: https://www.legislation.gov.uk/ukpga/2018/12/contents/enacted [Accessed 21/01/2021]

[69] Pew Research Center. *Reddit news users more likely to be male, young and digital in their news preferences*. Available from: https://www.journalism.org/2016/02/25/reddit-news-users-more-likely-to-be-male-young-and-digital-in-their-news-preferences/ [Accessed 10/06/2021]

[70] O. Kovaleva, A. Romanov, A. Rogers, A. Rumshisky. Revealing the Dark Secrets of BERT. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, November 2019. pp. 4365-4374. Available from: https://www.aclweb.org/anthology/D19-1445 [Accessed 14/06/2021]

[71] E.S. Jo, T. Gebru. Lessons from Archives: Strategies for Collecting Sociocultural Data in Machine Learning. *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 27th January 2020. pp. 306-316. Available from: https://dl.acm.org/doi/abs/10.1145/3351095.3372829 [Accessed 14/06/2021]

[72] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, Q.V. Le. XLNET: Generalized Autoregressive Pretraining for Language Understanding. *Advances in Neural Information Processing Systems*, 2019. Vol. 32. Available from: https://papers.nips.cc/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf [Accessed 14/06/2021]

[73] Mobilesquared. *WhatsApp users by country*. Available from: https://mobilesquared.co.uk/whatsapp-users-by-country/ [Accessed 15/06/2021]

[74] Goodwill Community Foundation, Inc. *What is an echo chamber?* Available from: https://edu.gcfglobal.org/en/digital-media-literacy/what-is-an-echo-chamber/1/ [Accessed 14/06/2021]