Social Network Analysis and Mining (2021) 11:69
https://doi.org/10.1007/s13278-021-00780-w

ORIGINAL ARTICLE

Progressive domain adaptation for detecting hate speech on social media with small training set and its application to COVID-19 concerned posts

Md Abul Bashar[1] · **Richi Nayak**[1] · **Khanh Luong**[1] · **Thirunavukarasu Balasubramaniam**[1]

Received: 8 November 2020 / Revised: 29 June 2021 / Accepted: 20 July 2021 / Published online: 29 July 2021

Abstract
In this world of information and experience era, microblogging sites have been commonly used to express people feelings including fear, panic, hate and abuse. Monitoring and control of abuse on social media, especially during pandemics such as COVID-19, can help in keeping the public sentiment and morale positive. Developing the fear and hate detection methods based on machine learning requires labelled data. However, obtaining the labelled data in suddenly changed circumstances as a pandemic is expensive and acquiring them in a short time is impractical. Related labelled hate data from other domains or previous incidents may be available. However, the predictive accuracy of these hate detection models decreases significantly if the data distribution of the target domain, where the prediction will be applied, is different. To address this problem, we propose a novel concept of unsupervised progressive domain adaptation based on a deep-learning language model generated through multiple text datasets. We showcase the efficacy of the proposed method in hate speech and fear detection on the tweets collection during COVID-19 where the labelled information is unavailable.

Keywords Domain adaptation · Hate speech · Small dataset · Text mining · Fear prediction

1 Introduction

Microblogging sites such as Twitter and Tumblr create interesting social networking structures by facilitating users to post contents and interact with each other through replies and reactions. As one of the biggest social media platforms, Twitter facilitates the posting of hundreds of millions of tweets daily. These tweets are available to be collected using Twitter's API. The enormous size of tweet data contains valuable insights that can be investigated in different ways such as tracking conversations, forming subgroups among posts (or users) as per common topics (or interests), or building classifiers to detect abuse.

When a large-scale incidence such as COVID-19 occurs, users start to use social media as an additional channel to follow official communication sources to create, share, validate, and disseminate crisis information [1] (Heverin and Zach 2012). With facilitating meaningful ideas and thoughts exchange, this infodemic also generates fear and panic due to unverified rumours and exaggerated claims and promotes abuse and racist forms of digital vigilantism [2] (Brindha et al. 2020).

Analysis of social media data can give us an immediate insight into the pandemic, which help in reducing costs to the economy over the long term and bringing harmony to the society (Bashar et al. 2020; Balasubramaniam et al. 2020). For keeping the public sentiment and morale positive, the problem of hate and harassment detection becomes crucial than ever. Understanding of information

✉ Md Abul Bashar
m1.bashar@qut.edu.au

Richi Nayak
r.nayak@qut.edu.au

Khanh Luong
khanh.luong@qut.edu.au

Thirunavukarasu Balasubramaniam
t.balasubramaniam@qut.edu.au

[1] School of Computer Science and Centre for Data Science, Queensland University of Technology, 2 George St, Brisbane City, QLD 4000, Australia

[1] https://www.nielsen.com/us/en/insights/article/2020/covid-19-tracking-the-impact-on-media-consumption/.

[2] https://theconversation.com/covid19-social-media-both-a-blessing-and-a-curse-during-coronavirus-pandemic-133596.

Fig. 1 Word cloud obtained from the random hate data (i.e. Source data)

Fig. 2 Word cloud obtained from the East Asian hate data (i.e. target data)

on COVID-19-related harassment and automatic detection of abuse will bring positive impact to society (Al-garadi et al. 2016).

These analyses such as the spread of fear, hate and racism require labelled data (Al-garadi et al. 2016). However, obtaining the labelled data in suddenly changed circumstances as the COVID-19 pandemic is expensive and acquiring them in a short time is impractical. Related labelled data from other domains or previous incidents is usually available (Al-garadi et al. 2016; Mohammad and Kiritchenko 2018). However, the predictive accuracy of these detection models decreases significantly if the data distribution of the target domain, where the prediction will be applied, is different (Long et al. 2015; Yosinski et al. 2014; Wang and Schneider 2014; Ghifary et al. 2014; Long et al. 2013; Baktashmotlagh et al. 2013; Gong et al. 2013; Pan et al. 2010; Xu et al. 2019; Han and Eisenstein 2019; Rietzler et al. 2019). This paper proposes an autonomous hate speech detection method for datasets where labelled data is not available, which is a requirement to build an effective predictive model.

Domain-adaptation-based transfer learning is a common approach to utilise data from related domains that may have different distributions (Chen et al. 2019; Tzeng et al. 2017; Long et al. 2015; Yosinski et al. 2014; Wang and Schneider 2014; Ghifary et al. 2014; Long et al. 2013; Baktashmotlagh et al. 2013; Gong et al. 2013; Pan et al. 2010; Xu et al. 2019; Han and Eisenstein 2019; Rietzler et al. 2019). The goal of domain adaptation is to transfer domain-invariant knowledge from the labelled source domain to the unlabelled target domain for bridging different domains with substantial distribution discrepancy (Pan and Yang 2009). A major challenge in domain adaptation for predictive models is data discrepancy in the domains (Long et al. 2015). For example, a hate speech detection model trained on common hate speech data obtained before the COVID-19 pandemic may not be able to effectively detect hate speech directed towards the East Asian Community during the pandemic. Word clouds generated from two such datasets shown in Figs. 1 and 2 highlight the variations in two distributions, e.g. vocabulary, term frequency and the use of language.

Significant progress has been made in computer vision on domain-invariant knowledge transfer using deep learning models. A common approach is to learn models from data that can bridge the source and target domains in an isomorphic latent feature space (Long et al. 2013, 2015; Yosinski et al. 2014; Wang and Schneider 2014; Ghifary et al. 2014; Baktashmotlagh et al. 2013; Gong et al. 2013; Pan et al. 2010). This research showed, in image data, deep neural networks can disentangle features with factors of variations underlying the data distribution and group features hierarchically according to their relatedness to invariant factors. This leads to the conjecture that features learned by deep representation have the potential for domain adaptation. However, research shows that the deep features transition occurs from general to specific along the network (Long et al. 2015). The degree of generalisation in the later layers depends on the specificity in the early (i.e. lower) layers. Because lower layers learn domain-specific features, the feature transferability drops significantly in higher layers if domain discrepancy increases significantly (Yosinski et al. 2014). Intuitively, features learned by pretraining a model in a general domain could increase generalisation along with the network and enhance transferability in the task-specific layers.

Latest computer vision research in transferability of deep features implies enhancing transferability in the task-specific layers of the deep neural network by explicitly reducing the domain discrepancy (Long et al. 2015). To obtain this objective in an unsupervised setting, we propose to progressively transfer learn a language model from the general domain to *source-specific* domain. This will allow the language model learning domain invariant features from general domain datasets and progressively include additional features from source-specific domain datasets as necessary. This is our first contribution in proposing an effective autonomous hate speech detection method for unlabelled or small labelled datasets.

Another issue in transfer learning-based classifiers is faced as follows. Disentangling the variational factors in higher layers of the network (through transfer learning to source domain) can enlarge the domain discrepancy between source and target as the model progressively adapts to the source domain. This is because deep feature representations progressively become more *compact* to source domain and mutually distinguishable (Glorot et al. 2011) from target domain. To address this issue, we propose to progressively transfer learning the language model from the source domain to target domain datasets. This will allow the language model to learn domain invariant features from the source to target domain datasets and adapt them to the target domain. This is our second contribution in proposing an effective autonomous hate speech detection method for unlabelled or small labelled datasets.

Limited research has been conducted in text processing for domain adaptation in deep learning (Xu et al. 2019; Han and Eisenstein 2019; Rietzler et al. 2019). Existing research use a huge model, BERT with 110 million parameters (Devlin et al. 2018). Pretraining and fine-tuning such a huge model and making inference with it is computationally expensive and impractical to deploy in a standard hardware environment.

In this paper, we propose a novel concept of progressive domain adaptation based on transfer learning of a language model through multiple text datasets. We propose to learn a deep feature representation that can capture necessary domain invariance and disentanglement for target domain adaptation by bridging between general domain, source domain and target domain. We apply the proposed method for hate speech detection during the COVID-19 pandemic on a large Twitter dataset where the labelled information is unavailable. We effectively use both labelled and unlabelled datasets in different stages of the learning process intending to discover tweets that contain signs of fear and hate.

To our best of knowledge, the idea of progressive domain adaptation through a language model has not been investigated before in machine learning. This method is highly applicable to pandemics like COVID-19 where there is no prior information available and the detection systems are critical for social goods and policy making. We propose this novel approach to deal with the problem of lack of labelled data and present a systematic study learning its effects and dependencies in text processing.

Firstly, we investigate whether adaptation can be sensitive to the domain knowledge of the pretraining corpus. Should a language model be adapted from a general domain to a source domain and then the source domain to a target domain to capture domain invariant features?

Secondly, there exist several neural network models for possible domain adaptation options. Because of different underlying assumptions and varying architectures, the right choice heavily depends on the problem at hand. We systematically instigate the suitability of deep learning models for this task. For example, a pretrained convolutional neural network (CNN) has been found effective in image classification (Sharif Razavian et al. 2014), whereas a language model-based pretrained model has been found effective for in-domain (i.e. trained and tested in the same domain) text classification (Howard and Ruder 2018; Bashar et al. 2020).

Thirdly, it requires a rigorous investigation as to what extent domain adaptation through a language model can learn domain invariant and disentangled features so that the classification model trained in the source domain works in the target domain.

Lastly, it is useful to have a statistical understanding of domain adaptation using language model modelled by a deep neural network (DNN) as NN is considered a black-box approach. A statistical understanding will help to comprehend why domain adaptation by language model in DNN is working, identify the potential application areas and investigate for future improvements. There are some studies on statistical understanding on NNs (Li and Gal 2017; Blundell et al. 2015; Gal 2016; MacKay 1992); however, there exists no study on domain adaptation by the language model.

2 Related works: hate speech detection

The spread of hate speech is proliferating in online social media contents. Different machine learning methods use different terms to express *hate* based on a specific case study and/or the use of baseline datasets. For example, authors in Davidson et al. (2017); Malmasi and Zampieri (2017) named their problem as hate speech detection considering the existence of offensive language since they use dataset HatebaseTwitter (Davidson et al. 2017) which includes three annotated classes: hate, offensive (but not hate) and neither; authors in Badjatiya et al. (2017) used VaseemA (Waseem and Hovy 2016) and VaseemB (Waseem 2016) which include racism, sexism and therefore referred their problem as racism detection; authors in Mozafari et al. (2020) use VaseemA and HatebaseTwitter datasets and call the problem as hate speech detection; authors in Bashar et al. (2020) use misogynistic data and named their problem as misogynistic detection; and authors in Founta et al. (2019) use a wide range of datasets including Cyberbullying, offensive, hate and Sarcasm datasets and named their problem as abuse detection.

Though the definitions and the use of these terms vary in the literature, most agree that all these problems include a process to detect tweets that use abusive or offensive language to target a person or a group (MacAvaney et al. 2019). These problems pose several common challenges such as subtleties in language, differing definitions of hate/abusive speech and limited labelled data for training and testing hate

speech detection models (MacAvaney et al. 2019). In this paper, we use this definition in the context of hate speech detection to COVID-19-related data which is unlabelled.

In general, these problems are dealt with machine learning-based text classification methods ranging from supervised to transfer learning, from traditional shallow machine learning to deep learning. Simple methods identify an instance as hate speech if it contains a potentially hateful keyword. However, these methods fail to detect hateful contents that are implicitly hateful and do not use certain keywords (Bashar et al. 2020, 2018). Moreover, some of these keywords may appear sarcastically and are not always hateful (e.g. swine, trash, etc.). These methods result in detecting many false positives (MacAvaney et al. 2019).

2.1 Traditional machine learning algorithms

The first and long-known family uses traditional supervised algorithms including Support Vector Machine (SVM), Logistic Regression, Random Forest, or ensemble framework. These methods manually engineer the features, i.e. using terms or phrases (n-grams) to build the classifier based on labelled data (MacAvaney et al. 2019). SVM is used to train and decide a hateful tweet with the presence of profane but not hateful content (Malmasi and Zampieri 2017). Logistic regression was selected as the final modal to identify tweets as hateful, offensive or neither after comparing with algorithms such as Naïve Bayes, Decision trees, Random Forest and SVMs (Davidson et al. 2017). Authors in Rajalakshmi and Reddy (2019) designed two models using logistic regression and Random Forest by making use of different weighting methods including TF-IDF, Mutual Information and Chi-square to detect hate and offensive German and Hindi tweets.

2.2 Deep learning algorithms

Another family of methods based on deep learning architectures have shown promising results in the presence of large labelled data since they can learn hidden nonlinear patterns embedded in tweets. They use a complex multiple layers model and integrate complex linguistic contexts using word embeddings (e.g. Word2Vec Mikolov et al. 2013 or the whole sequence of words Kuncoro et al. 2018) to capture semantics (Devlin et al. 2018; Yang et al. 2018). These methods vary by using different architectures or making use of different input text/meta data (Founta et al. 2019). Different architectures such as CNN, LSTM (Long Short-Term Memory, a special case of Recurrent Neural Networks (RNNs)) and fastText[3] were used with different types of features to detect hateful tweets based on labelled racism and sexism tweets (Badjatiya et al. 2017). Solving the same problem, a CNN model was built with features embeddings such as one-hot encoded n-gram vectors and word embeddings (Gambäck and Sikdar 2017), or with characters and word level inputs (Park and Fung 2017).

However, these techniques are only effective when the model is trained and tested in the same domain. Their performance significantly drops when trained in one domain and tested in another domain with varying data distribution. *In this paper, we propose a novel domain adaptation method for the deep learning-based classifier to address this situation when there is no labelled target data available for building a hate speech detection classifier.*

2.3 Transfer learning algorithms

The most relevant to the proposed method is the emerging family of methods based on transfer learning. Becoming common in computer vision, it is a learning technique that utilises knowledge that has been learned before in other tasks or domains. Generally, these methods follow a two-step process: the first step is to pretrain the model using an available dataset and available language model architecture, and the second step is to fine-tune the model to the target domain or task.

Depending on the availability of labelled data in both source and target domains, these methods can be divided into different groups such as self-taught learning, multi-task learning, domain adaptation or unsupervised transfer learning. *The problem considered in this paper falls into the domain adaption problem where we have limited labelled data available in source domain and only unlabelled data is available in target domain.*

In computer vision, models learn domain-invariant knowledge from data that can bridge the source and target domains in an isomorphic latent feature space (Chen et al. 2019; Tzeng et al. 2017; Long et al. 2015; Yosinski et al. 2014; Wang and Schneider 2014; Ghifary et al. 2014; Long et al. 2013; Baktashmotlagh et al. 2013; Gong et al. 2013; Pan et al. 2010). For example, Adversarial Discriminative Domain Adaptation (ADDA) Tzeng et al. (2017) has an embedding part and a classification part. The embedding part first learns a discriminative representation using the labels in the source domain. The target data is then used to fine-tune the embedding part through a domain-adversarial loss. This process of ADDA is further improved in Progressive Feature Alignment Network (PFAN) (Chen et al. 2019) that combines three losses, namely Adaptive Prototype Alignment loss, Class Discrimination loss and Domain Discrimination loss to fine-tune the embedding network to the target data. However, these models require a large amount of labelled data in the

[3] https://fasttext.cc.

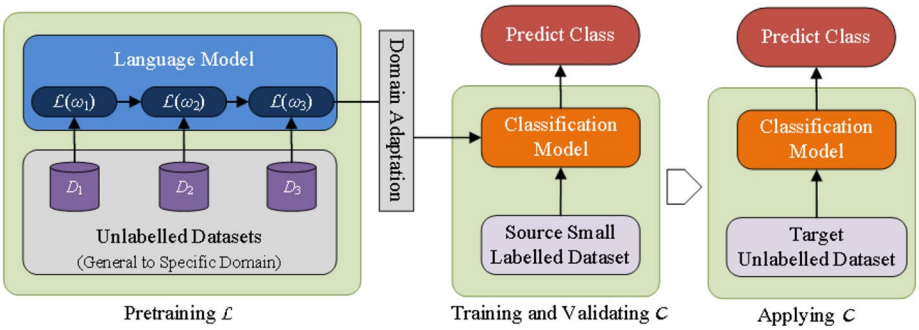

Fig. 3 Progressive transfer learning for domain adaptation

source domain. On the other hand, the proposed method does not need labelled data in its embedding part as it uses a language model for learning the word embedding from the source domain and fine-tuning the embedding to the target domain. Our experimental results in this research confirm that if the source domain does not have a large amount of labelled data, these models such as ADDA and PFAN cannot perform well.

Recently pretrained language models have shown to be an effective transfer learning for many downstream text processing tasks (Bashar et al. 2020; Howard and Ruder 2018; Devlin et al. 2018). In particular, a language model-based transfer learning was used for detecting misogynistic tweets when there is a small set of labelled data available for training the classifier (Bashar et al. 2020). A handful of methods exist that transfer knowledge from a different source task. For example, the Multi-Granularity Alignment Networks (MGAN) (Li et al. 2019) model is proposed to transfer knowledge from an aspect-category classification task. Knowledge is transferred from a document-level sentiment classification task by reusing the weights of an LSTM network (He et al. 2018). More commonly, the BERT language model is fine-tuned to transfer knowledge across different domains (Xu et al. 2019; Han and Eisenstein 2019; Rietzler et al. 2019).

BERT is a huge language model with 110 million parameters (Devlin et al. 2018) that predicts *masked* words instead of a *next* word in a given sequence. Pretraining, fine-tuning and making inference are computationally very expensive for such a huge model and impractical to deploy in a standard hardware environment owned by most organisations except the technology Giants like Google, Facebook, Amazon or OpenAI. BERT is pretrained using a huge dataset. In contrast, we propose a much smaller language model based on LSTM with 24 million parameters. Such a smaller model allows us to pretrain it with a smaller dataset. However, a small dataset cannot cover the multitude of domains often required by domain adaptation. Therefore, we propose a novel technique of progressive domain adaptation with multiple datasets that cover multiple domains for effective domain adaptation.

To the best of our knowledge, ours is the first work proposing the progressive domain adaptation through a language model and validating the model in the task of hate speech detection effectively.

3 Proposed method

In this section, we present the proposed methodology for unsupervised domain adaptation by progressive transfer learning of language model. We consider unsupervised domain adaptation with a limited labelled source data for building a text classifier for unlabelled target data. Figure 3 shows the overview of the proposed method. We use progressive transfer learning of language model for linking context (e.g. how language is used) and features (e.g. words) by capturing necessary domain invariance and disentanglement by bridging among general, source and target domains.

Suppose there exist a limited labelled source dataset $D^s = \{X_i^s, Y_i^s\}$, a target unlabelled dataset $D^t = \{X_i^t\}$ and a total of N_r unlabelled datasets $\{D_k^r\}_{k=1}^{N_r} = \{X_i^{rk}\}_{k=1}^{N_r}$ related to D^s or D^t. Assume that the data distribution of source and target domain is different, i.e. $P_s(X_i^s, Y_i^s) \neq P_t(X_i^t, Y_i^t)$ where Y_i^t are predicted labels of target samples. It can be said that P_t is *changed* from P_s by some *domain shift*. The objective of unsupervised domain adaptation is to obtain a model $\mathcal{F} : X \rightarrow Y$ that can predict corresponding labels $\{Y_i^t\}$ for $\{X_i^t\}$ given $\{X_i^s, Y_i^s\}$, $\{X_i^t\}$ and $\{X_i^{rk}\}_{k=1}^{N_r}$ during the training as inputs.

We decompose the mapping $\mathcal{F} : X \rightarrow Y$ into two parts. First, the input X_i^s is mapped to a $\mathbf{D}$-dimensional latent deep feature vector $\mathbf{X}_i \in \mathbb{R}^{\mathbf{D}}$ by a language model $\mathcal{L}$, i.e. $\mathbf{X}_i = \mathcal{L}(X_i;\omega)$ where ω is the parameter set in the language model. Then, the feature vector $\mathbf{X}_i$ is mapped to label Y_i by a classification model $\mathcal{C}$, i.e. $Y_i = \mathcal{C}(\mathbf{X}_i;\theta)$ where θ is the parameter set in the classification model.

Our goal is to make the features $\mathbf{X}_i$ domain-invariant between source D_s and target D_t. That is, we want to make the distribution $P_s(\mathbf{X}_i) = \{\mathcal{L}(X_i;\omega_s)|X_i \sim P_s(X_i)\}$ and $P_t(\mathbf{X}_i) = \{\mathcal{L}(X_i;\omega_t)|X_i \sim P_t(X_i)\}$ is similar, where ω_s is the set of parameters learned in the source domain and ω_t is the

set of parameters learned in the target domain. According to the covariate shift assumption, this would make the label prediction accuracy on the target domain to be the same as on the source domain (Shimodaira 2000).

The dissimilarity calculation between distributions $P_s(\mathbf{X}_i)$ and $P_t(\mathbf{X}_i)$ is non-trivial as $\mathbf{X}$ is high-dimensional and distributions are constantly changing as the learning progresses (Ganin and Lempitsky 2015). One way to estimate the dissimilarity is to look at the loss of the Language Model $\mathcal{L}$ in cross-domain, i.e. trained in source domain and evaluated in target domain or vice versa. This can be true if the parameters ω of the Language model have been optimally trained to generate texts, i.e. predicting $(n+1)^{\text{th}}$ word given previous n words.

To achieve the domain invariant objective, we propose progressively transfer learning through $\{D_k^r\}_{k=1}^{N_r}$ datasets that progressively relate to the target domain from a source domain. The goal is to achieve a smooth probability distribution of features from the source to target domains. The progressive transfer learning is described in detail in the following subsection. We then fine-tune the language model $\mathcal{L}$ in the source domain D^s to use the latent deep features for training a classifier $\mathcal{C}$ with the D^s dataset. After training, the classifier is applied to target domain D^t.

In summary, we want the parameters of both the language model and the classifier to be optimised to minimise the empirical loss for target domain samples. This requires the discriminativeness of features in $\mathcal{C}$ to classify the instances and generalisation in $\mathcal{L}$ so that the learned features can be effective in the target domain for the classification task. In other words, we want to implicitly reduce the loss $L(\mathcal{F}) = L(\mathcal{L}) + L(\mathcal{C})$.

3.1 Language model

Let $X = (x_1, \dots x_n)$ be a feature vector representing an instance or sample. A language model $\mathcal{L} = p(x_j|x_1 \dots x_{j-1})$ seeks to predict the probability of observing the jth feature x_j, given the previous $(j-1)$ features $(x_1 \dots x_{j-1})$. $\prod_{j=1}^{n} p(x_j|x_1 \dots x_{j-1}) = p(x_1, \dots, x_n) = p(X)$ can be interpreted as the probability of observing an instance (e.g. a sequence or a sentence) in a dataset. However, it is computationally difficult to estimate $\prod_{j=1}^{n} p(x_j|x_1 \dots x_{j-1})$ (Bashar et al. 2020). A simple estimation for this can be:

$$p(x_j|x_1, \dots, x_{j-1}) = \frac{count(x_1, \dots, x_{j-1}, x_j)}{count(x_1, \dots, x_{j-1})}$$

Nonetheless, observing enough data (in order to obtain realistic counts for any sequence of j features for any non-trivial value of j) from a dataset is unrealistic (Bashar et al. 2020). Therefore, the Markov assumption can be used to address this problem (Hausman and Woodward 1999). It assumes that the probability of observing a feature x_j at a given position j of the sequence is only dependent on the features observed in the previous $(j-1, \dots, j-c)$ positions, and independent of the features observed in all of the positions before $j-c$.

$$\begin{aligned} p(x_j|x_1, \dots, x_{j-1}) &\equiv p(x_j|x_{j-c}, \dots, x_{j-1}) \\ p(x_n, \dots, x_1) &\equiv \prod_{j=1}^{n} p(x_j|x_{j-c}, \dots, x_{j-1}) \end{aligned} \tag{1}$$

Adhering to this, a Word2Vec model uses a sliding window S of size c over the corpus $\{X_i\}$ to learn word embeddings (Mikolov et al. 2013). Within the window S, the Continuous Bag-of-Words (CBOW) model of Word2Vec uses context (surrounding words) to predict a target word, and the skip-gram model of Word2Vec uses a target word to predict a context. For the CBOW model, Eq. 1 can be written as,

$$\begin{aligned} ln\ p(x_n, \dots, x_1) &\equiv \sum_{j=1}^{n} ln\ p(x_j|x_{j-c}, \dots, x_{j-1}) \\ &\equiv \sum_{j=1}^{n-c+1} \sum_{x_k \in S_j} ln\ p(x_k|S_j - \{x_k\}) \end{aligned} \tag{2}$$

where S_j represents the starting position of sliding window S at jth word (or feature). The objective of Word2Vec is to learn model parameters ϑ (a.k.a. word embeddings or word vectors of N words) from a large unlabelled dataset to maximise $p(x_N, \dots, x_1)$ over the dataset. That is,

$$\arg\max_{\vartheta} \sum_{i=1}^{N-c+1} \sum_{x_j \in S_i} ln\ p(x_j|S_i - \{x_j\}, \vartheta)$$

The learned N word vectors (ϑ) can be viewed as the approximate distributed representation of $p(x_N, \dots, x_1)$ over the dataset.

Even though word embeddings have been used in a myriad of applications (Mikolov et al. 2013; Wang et al. 2019), it ignores two important characteristics of a given sequence: (i) the order of features in the sliding window S; and (ii) nonlinear and hierarchical interactions between features. Since Word2Vec uses a single hidden layer, it can capture only the linear interaction between features. However, features of a sequence (e.g. a sentence in a natural language) can have many levels of nonlinear hierarchical interactions (Mnih et al. 2009). An effective language model $\mathcal{L}$ should capture the order of features and their nonlinear interactions. This capacity allows a language model to encode the complexity of a language such as grammatical structure as well as to distil a fair amount of knowledge from the corpus (Jozefowicz et al. 2016).

To capture the order of features and their nonlinear and hierarchical interactions, an RNN or its variants such as

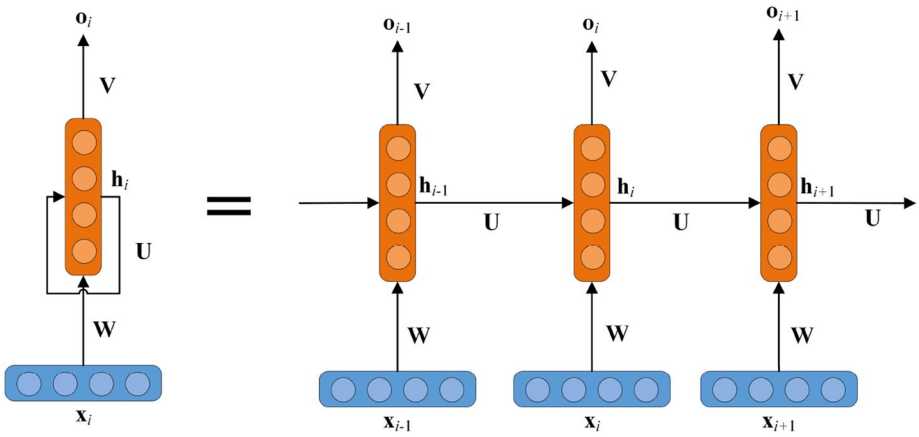

Fig. 4 A single-layered RNN model

LSTM can be used (Mikolov et al. 2010; Jozefowicz et al. 2016). Given a sequence of features, an RNN recurrently processes each feature and uses multiple hidden layers to capture the order of features and their nonlinear and hierarchical interactions. A simple RNN, as shown in Fig. 4, is constructed by repeatedly applying a function f_h that generates a hidden state $\mathbf{h}_j$ for jth feature x_j represented with vector $\mathbf{x}_j$, i.e.

$$\mathbf{h}_j = f_h(\mathbf{x}_j, \mathbf{h}_{j-1}) = \varphi(\mathbf{x}_j\mathbf{W} + \mathbf{h}_{j-1}\mathbf{U} + \mathbf{b})$$

where $\mathbf{W}$ is the parameter (weight) matrix for *input to hidden* layer for jth feature, $\mathbf{U}$ is the parameter matrix for hidden layer of $(j-1)$th feature to hidden layer of jth feature, $\mathbf{b}$ is bias vector, and φ is a (nonlinear) activation function such as tanh or Rectified Linear Unit (ReLU).

The hidden state is used to derive a vector of probabilities representing the network's prediction of the subsequent feature in the sequence. The network aims to minimise the loss calculated based on the vector of probabilities and the actual next feature. In simple words, the context of all previous features in the sequence is encoded within the parameters $\boldsymbol{\omega}$ of the network and the probability of getting the next word is distributed over the vocabulary using a Softmax function (Jozefowicz et al. 2016). The model output $\mathbf{o}$ can be defined as follows.

$$\mathbf{o}_j = f_o(\mathbf{h}_j) = \sigma(\mathbf{h}_j\mathbf{V}) + \mathbf{a}$$

where $\mathbf{V}$ is the parameter matrix for a *hidden layer to output* for jth feature, $\mathbf{a}$ is the bias vector, and σ is a softmax function used to convert the result into a probability distribution over vocabulary. LSTM (Hochreiter and Schmidhuber 1997) uses a gating mechanism to ensure proper propagation of information through many steps of a sequence to retain long-term dependencies.

It can be noted that a RNN/LSTM can approximate the Language model $\mathcal{L}$ by approximating joint probabilities over the feature sequences:

$$\prod_{j=1}^{n} p(x_j|x_1 \dots x_{j-1}, \boldsymbol{\omega}) \approx p(x_1, \dots, x_n) \quad = p(X) \tag{3}$$

In this research, we use the hidden state to obtain the $\mathbf{D}$ -dimensional latent deep feature vector $\mathbf{X}_i \in \mathbb{R}^{\mathbf{D}}$ for the sequence X.

3.2 Progressive transfer learning for domain adaptation

When a model learns $p(X)$ from a dataset D, the learned probability distribution depends on the corpus D. In other words, $p(X)$ is conditioned on the corpus D, i.e. $p(X, D) = p(X|D)p(D)$. As $p(\mathbf{X}) = \{p(\mathcal{L}(X;\omega))|X \sim p(X)\}$, distribution $p(\mathbf{X})$ depends on the corpus D.

The objective is make the latent features $\mathbf{X} = \mathcal{L}(X)$ to be invariant for source D^s and target D^t domains. The problem of making $\mathbf{X}$ domain invariant through transfer learning for a small source dataset is not well studied. When the source domain dataset D^s is small, using existing techniques as in computer vision such as Ganin and Lempitsky (2015), Long et al. (2015), Hoffman et al. (2018), and Xu et al. (2019) is not feasible. A larger dataset can more closely approximate the population (Banko and Brill 2001). However, if the dataset is small, we run the risk of learning properties that are unusual just by chance. The smaller the dataset, the higher the risk.

The traditional transfer learning approach in text processing that learns $p(X)$ from a single dataset (Jozefowicz et al. 2016; Merity et al. 2017; Melis er al. 2017) can be benefited using a large dataset that covers multiple related domains for achieving domain invariance. However, estimating $p(X)$ using an RNN/LSTM model on a huge dataset that covers the multitude of domains can be very expensive in terms of required computation and memory (Bradbury et al. 2016). Besides, it can learn irrelevant and misleading relationships

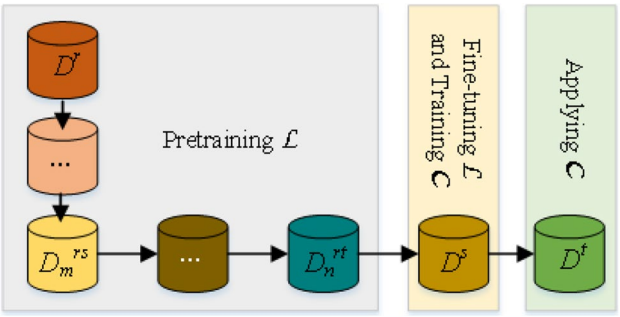

Fig. 5 Progressive domain adaptation of language model $\mathcal{L}$

in data due to interactions between different domains in a single corpus (Bashar et al. 2020).

To address these issues, we propose a progressive domain adaptation method based on transfer learning of language model $\mathcal{L}$ as shown in Fig. 5. This can incorporate knowledge from multiple datasets to make $\mathbf{X}$ domain invariant for source domain D^s and target domain D^t while discriminative for the task-specific classification in $\mathcal{C}$.

Let there be m number of datasets or corpora $\{D^s, D^t, \{D_k^r\}_{k=1}^{N_r}\}$ from which the multi-domain knowledge can be gained.

$$\begin{aligned} &p(\mathbf{X}, D_1, \ldots, D_m) \\ &\quad = p(\mathbf{X}|D_1, \ldots, D_m)p(D_1, \ldots, D_m) \\ &\quad = p(\mathbf{X}|D_1, \ldots, D_m)p(D_1|D_2, \ldots, D_m) \\ &\qquad \ldots p(D_{m-1}|D_m)p(D_m) \end{aligned}$$

If we assume the datasets are independent of each other, applying the Naive Bayes assumption, we can write

$$\begin{aligned} &p(\mathbf{X}|D_1, \ldots, D_m)p(D_1|D_2, \ldots, D_m) \ldots p(D_{m-1}|D_m)p(D_m) \\ &\quad \propto p(\mathbf{X}|D_1, \ldots, D_m)p(D_1)p(D_2) \ldots p(D_m) \\ &\quad \propto p(\mathbf{X}|D_1)p(\mathbf{X}|D_2) \ldots p(\mathbf{X}|D_m)p(D_1)p(D_2) \ldots p(D_m) \\ &\quad = p(\mathbf{X}|D_1)p(D_1)p(\mathbf{X}|D_2)p(D_2) \ldots p(\mathbf{X}|D_m)p(D_m) \\ &\quad = \prod_{k=1}^{m} p(\mathbf{X}|D_k)p(D_k) \end{aligned}$$

A RNN/LSTM model built on corpus D_k to learn $p(\mathbf{X}|D_k)p(D_k)$ will have its parameters ω_k. It can be expressed as follows.

$$\begin{aligned} \prod_{k=1}^{m} p(\mathbf{X}|D_k)p(D_k) &\approx \prod_{k=1}^{m} p(\mathbf{X}|D_k, \omega_k)p(D_k, \omega_k) \\ &= \prod_{k=1}^{m} p(\mathbf{X}|D_k, \omega_k)p(\omega_k|D_k)p(D_k) \end{aligned}$$

If the same language model $\mathcal{L}$ is sequentially built from the given m datasets, parameters ω_i learned on i^{th} dataset will only depend on the parameters ω_{i-1} learned on the $(i-1)^{th}$ dataset, applying the Markov assumption (Hausman and Woodward 1999).

$$\begin{aligned} &\prod_{k=1}^{m} p(\mathbf{X}|D_k, \omega_k)p(\omega_k|D_k)p(D_k) \\ &\quad \approx \prod_{k=1}^{m} p(\mathbf{X}|D_k, \omega_k)p(\omega_k|D_k, \omega_{k-1})p(D_k) \end{aligned}$$

Here ω_0 is the initial weight that may be assigned randomly. Assuming the same probability (or uncertainty) for each dataset, domain adaptation can be expressed as follows.

$$\begin{aligned} p(\mathbf{X}, D_1, \ldots, D_n) &\approx \prod_{k=1}^{m} p(\mathbf{X}|D_k, \omega_k)p(\omega_k|D_k, \omega_{k-1})p(D_k) \\ &= \prod_{k=1}^{m} p(\mathbf{X}|D_k, \omega_k)p(\omega_k|D_k, \omega_{k-1}) \\ &\propto \sum_{k=1}^{m} ln\big(p(\mathbf{X}|D_k, \omega_k)p(\omega_k|D_k, \omega_{k-1})\big) \end{aligned} \tag{4}$$

The following observations can be made based on Eq. 4. (1) Each dataset D_k relevant to the source or target domain can reduce uncertainty. This reinforces the previous findings based on word embedding that if word vectors of Word-2Vec are pretrained using a corpus relevant to the target task domain, performance of the target task is significantly improved (Bashar et al. 2018). (2) Pre-training of RNN/LSTM for $\mathcal{L}$ should be done by the order of the dataset of a general population distribution to the dataset of specific population distribution because the parameter vector ω_k depends on ω_{k-1}. For example, we can approximate the population distribution of Queensland (i.e. specific) from that of Australia (i.e. general) but the opposite is not true. (3) It may be difficult to decide which one of the source and target datasets is more general. We want to train our classifier using source labelled dataset D^s and want to apply it to the target domain D^t. This means parameters ω_t of $\mathcal{L}(\mathbf{X}|\mathbf{D^t}, \boldsymbol{\omega_t})$ will depend on parameters ω_s of $\mathcal{L}(\mathbf{X}|\mathbf{D^s}, \boldsymbol{\omega_s})$. Therefore, $\mathcal{L}$ will need to be trained from source domain-specific datasets to the target domain-specific dataset.

3.3 Classifier domain adaptation

Disentanglement of variational factors mainly happens in the classifier training. The goal of classifier domain adaptation is to transform deep features in a fashion that changes only necessary properties of the underlying deep state while leaving all other properties invariant. This is supposed to give exploitable structure to any kind of data (Higgins et al. 2018).

Let D^s be a small dataset that contains N_f features and N_c classes. $X = (x_1, \ldots x_n)$ is a feature vector representing

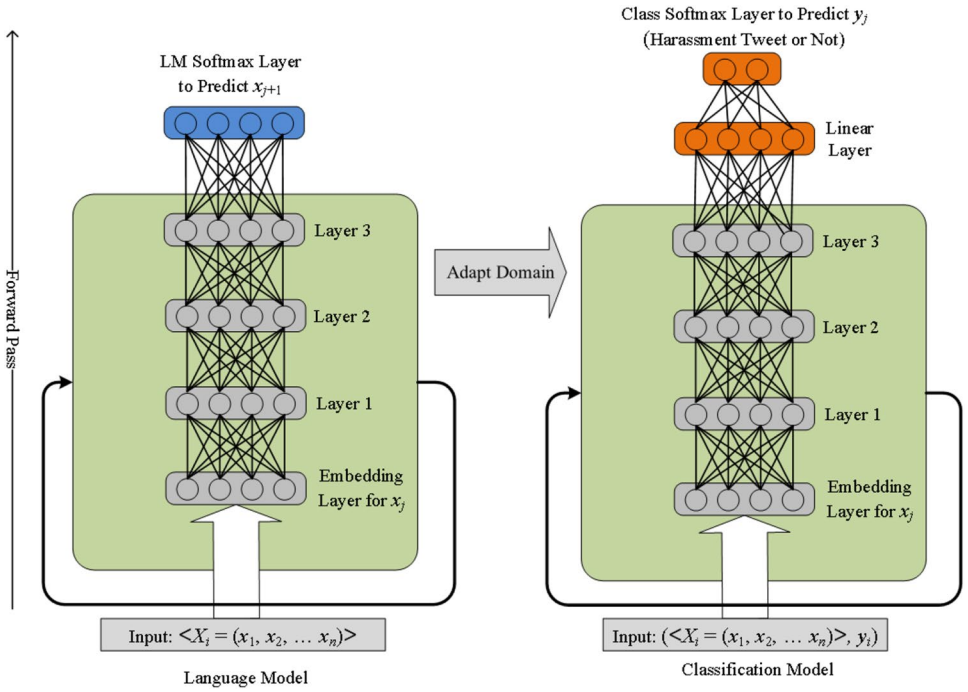

Fig. 6 Model architecture of $\mathcal{L}$ and $\mathcal{C}$ for progressive domain adaptation

an instance in D^s. Let C be a set of N_c classes. The classification task is to assign an instance to a class C_l based on the feature vector X.

In this research, we use LSTM for both language model $\mathcal{L}$ and classifier $\mathcal{C}$. Figure 6 shows the proposed architecture. When a LSTM model is trained to assign X to C_l, first it learns the latent feature vector $\mathbf{X}$ for X. This type of classification models learns a joint probability distribution $p(C_l, \mathbf{X}, \theta)$ that can be written as,

$$p(C_l, \mathbf{X}, \theta) = P(C_l|\mathbf{X}, \theta)p(\mathbf{X}) \tag{5}$$

where θ is the set of model parameters, $P(C_l|\mathbf{X}, \theta)$ is the discriminative probability leaned by the model and $p(\mathbf{X})$ is the prior probability.

In the *Classification Model* of Fig. 6, earlier layers preceding the *Linear Layer* learn $p(\mathbf{X})$ that summarises X prior to learning discriminative probability. The *Linear Layer* and the *Class Softmax* layer together learn the discriminative probability $P(C_l|\mathbf{X}, \theta)$. Note that $p(\mathbf{X})$ in Eq. 5 can regularise $p(C_l|\mathbf{X}, \theta)$ as pointed out in our prior work (Bashar et al. 2020).

In transfer learning, as $p(\mathbf{X})$ does not depend on class label C_l, $p(\mathbf{X})$ is learned from some external unlabelled datasets and fine-tuned during classification (Bashar et al. 2020). We argue that if $p(\mathbf{X})$ can be learned domain invariant for source domain D^s and target domain D^t, then $p(\mathbf{X})$ can adapt $P(C_l|\mathbf{X}, \theta)$ for target domain D^t when trained in source domain D^s.

In Sect. 3.2, we described how to learn $\mathbf{X}$ as domain invariant to represent X using progressive transfer learning of $\mathcal{L}$ in m number of unlabelled datasets. We use the *Language Model* architecture in Fig. 6 for progressive transfer learning of our language model for learning $p(\mathbf{X})$ as domain invariant.

In both language and classification models, each layer preceding *Softmax* and *Liner Layer* learns hierarchically general to specific *latent* feature vectors $\langle \mathbf{X}^1, \ldots, \mathbf{X}^n \rangle$ for feature vector X.

As we discussed in Sect. 3.2, we use progressive domain adaptation for learning $\mathbf{X}$ in a domain invariant way. Fine-tuning during classifier training can abruptly change $\mathbf{X}$. Therefore, to keep $\mathbf{X}$ domain invariant in the classification model, we keep earlier layers frozen throughout the training, i.e. $\langle \mathbf{X}^1, \ldots, \mathbf{X}^{n-2} \rangle >$ are not updated during the classifier training. After classifier $\mathcal{C}$ has been trained and validated on source dataset D^s, we apply $\mathcal{C}$ on target dataset D^t. In dataset D^t no parameter $\mathcal{C}$ is updated, i.e. no learning is done in D^t.

Next we empirically analyse how much $p(C_l|\mathbf{X}, \theta)$ can be adapted to target domain using $p(\mathbf{X})$ leaned by progressive transfer learning of $\mathcal{L}$ when a small number of data is available in source domain $D^s = \{X_i^s, Y_i^s\}$.

4 Empirical evaluation

We name the proposed LSTM-based progressive domain adapted classification model as LSTM-DA. The primary objectives of LSTM-DA evaluation are to show the effectiveness of progressive domain adaptation when the target domain has no labelled data and the source domain has a small set of labelled data. We investigate the followings: (a) sensitivity of domain adaptation to the domain knowledge of pretraining datasets; (b) sensitivity of progressive domain adaptation to the order of pretraining datasets (or domains); (c) effectiveness of different models in domain adaptation (trained in the source domain and applied in target domain); (d) comparison of the progressively domain adapted model with other state-of-the-art models when trained and applied in the same domain (i.e. in-domain performance). All experiments were conducted to achieve the best accuracy performance in detecting hate speech in tweets.

4.1 Data collection

We use several datasets for tuning a language model, building a classifier and evaluating the performance of the classifier for hate detection.

4.1.1 Target dataset: east Asia hate dataset (EAHD)

The dataset EAHD (Vidgen et al. 2020) is used as the target dataset D^t. It represents the domain of COVID-19 East Asia hate, collected between 1st January and 17th March 2020. EAHD includes a total number of 20,000 tweets labelled as to whether a tweet is East Asian relevant and, if so, what is the stance (Very Negative, Negative, Neutral, Positive and Very Positive). A total of 3898 instances are labelled positive (i.e. very negative or negative stance towards East Asian people). To remove the skewness in the data class, we randomly selected a total of 3898 instances labelled as Neutral, Positive and Very Positive. In our experiments, we use this subset of data containing a total of 7,796 instances. Labels are only used for evaluation.

4.1.2 Source labelled dataset: general hate dataset (GHD)

The dataset GHD, collected from Kaggle[4], is used as the source dataset D^s. GHD includes the tweets for the year 2018 before the COVID-19 pandemic broke. GHD is in the domain of general hate before COVID-19, e.g. hate against Muslims, black people, white people, women, etc. This dataset has a total of 31,962 tweets out of which 2,242 instances are positive (i.e. hate). After stratification, a total number of 4,484 instances remained in the subset of data for experimental evaluation.

4.1.3 Pretraining datasets for domain adaptation

1. Dataset D_1: Wiki103 This is our general domain dataset. The goal of using this corpus is to capture the general properties of the English language. We pretrain the Language model on Wikitext-103 that contains 28,595 verified good quality and featured Wikipedia articles and 103 million words (Merity et al. 2016). After pretraining the language model on D_1, we approximate the probability distribution $p(\mathbf{X}|D_1, \boldsymbol{\omega}_1)$.
2. Dataset D_2: Random Global Tweets (RGT) The goal of using this corpus is to bridge the data distribution between the general domain dataset D_1 and the source domain D^s (i.e. general hate dataset GHD). D_2 is needed because the source domain dataset D^s is small and likely has a different distribution than the general corpus D_1. Dataset D_2 should be chosen such that it is relevant to D^s. D_2 contains 16.28 million random tweets collected in 2018 before the COVID-19 pandemic started. The Twitter Stream Application Programming Interface (API) was used in collecting this dataset.

   As $p(\mathbf{X}|D_2, \omega_2)p(\omega_2|D_2, \omega_1)$ is approximated on D_2, the parameter set ω_2 can be considered tuned with D_2 and ω_1. D_2 may not be the complete subset of D_1, i.e. D_2 may contain some exclusive information other than D_1. This means $p(\mathbf{X}|D_2, \omega_2)p(\omega_2|D_2, \omega_1)$ is more specific and less uncertain than $p(\mathbf{X}|D_1, \boldsymbol{\omega}_1)$ in relation to source domain D^s. We propose using *discriminative fine-tuning* (tune each layer of LSTM with different learning rates) and *slanted triangular learning rates* (first rapidly increases the learning rate and then slowly decays) Howard and Ruder (2018) for fine-tuning the $\mathcal{L}$ with D_2.
3. Dataset D_3: COVID-19 Australian Sphere Tweets (CAST) The goal of this corpus is to capture the target domain D^t relevant specific factors. D_3 is needed because the target domain dataset D^t likely has a different distribution than the source domain-relevant dataset D_2. D_3 should be chosen such that it is relevant to D^t. D_3 contains twitter conversation in the Australian Sphere on COVID-19 from 27th November 2019 when the first break out occurred in China to 7th September 2020. The data collection is done via the QUT facility of Digital Observatory[5] using the Twitter Stream API. The dataset consists of 6.8 million tweets. Every tweet in the dataset contains or uses as a hashtag at least one of the following keywords: coronavirus, covid19, covid-19,

[4] https://www.kaggle.com/vkrahul/twitter-hate-speech.

[5] https://https://www.qut.edu.au/institute-for-future-environments/facilities/digital-observatory.

covid_19, coronovirusoutbreak, covid2019, covid, and coronaoutbreak.

Once fine-tuning of $\mathcal{L}$ is done on D_3, we get an approximation for $p(\mathbf{X}, D_1, D_2, D_3) \approx p(\mathbf{X}, \omega)$ which is an approximation for $p(\mathbf{X})$. Similar to D_2, we use *discriminative fine-tuning* and *slanted triangular learning rates* for fine-tuning the $\mathcal{L}$ with D_3.

4.2 Evaluation measures ad experimental setting

We used six standard classification evaluation measures (Bashar et al. 2020): Accuracy (Ac), Precision (Pr), Recall (Re), F_1 Score (F_1), Cohen Kappa (CK) and Area Under Curve (AUC). We also report True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) values.

For training LSTM-DA, we use *concat pooling* and *gradual unfreezing* techniques as in Howard and Ruder (2018), Bashar et al. (2020). For building the language model $\mathcal{L}$, we use the state-of-the-art AWD-LSTM Merity et al. (2017) which is a standard LSTM with various tuned dropout hyper parameters. The architecture and hyper parameters of $\mathcal{L}$ are the same as that used in (Howard and Ruder 2018; Bashar et al. 2020). We use ReLU activations for the intermediate layers and the softmax activation at the last layer that outputs probability distributions over the target vocabulary (for $\mathcal{L}$) or classes (for classifier $\mathcal{C}$). Hyperparameters are tuned using cross-validation. We used Python Machine Learning Library PyTorch[6] to implement this model. Coding was done using Jupyter Notebook[7] and executed on a Linux machine[8].

4.3 Benchmarking models

We implemented 16 baseline models to compare the performance of the proposed progressive domain adaptation based on the LSTM language model (named as LSTM-DA in experiments).

- Deep neural network models adapted to target domain by Word2Vec include (1) LSTM adapted by Word2Vec (LSTM-W) Hochreiter and Schmidhuber (1997) and (2) CNN adapted by Word2Vec (CNN-W) (Bashar et al. 2018). LSTM-W has 100 units, 50% dropout, binary cross-entropy loss function, Adam optimiser and sigmoid activation. The hyperparameters of CNN-W are set as in Bashar et al. (2018). The word vectors have 200 dimensions and are pretrained on dataset D_3: COVID-19 Australian Sphere Tweets (CAST). A Continuous Bag-of-Words Word2vec (Mikolov et al. 2013) model is used in pretraining while the minimum count for word is set to 100.
- Deep neural network models adapted to target domain by adapting the embedding network include (1) adversarial discriminative domain adaptation (ADDA) (Tzeng et al. 2017) and (2) progressive feature alignment network (PFAN) (Chen et al. 2019). ADDA has an embedding part and a classification part. The embedding part first learns a discriminative representation using the labels in the source domain. The target data is then used to fine-tune the embedding part through a domain-adversarial loss. Originally, ADDA was proposed for domain adaption in computer vision, where they used ResNet-50 as the base model due to its suitability in computer vision. However, ResNet-50 is not suitable for text data. Therefore, we have replaced ResNet-50 with two separate networks (1) an LSTM that we call ADDA-LSTM and (2) a CNN that we call ADDA-CNN, because experiments show that LSTM and CNN are suitable for text data (Bashar et al. 2018, 2020; Bashar and Nayak 2019, 2021). PFAN can be considered as a more sophisticated version of ADDA. PFAN combines three losses, namely adaptive prototype alignment loss, class discrimination loss and domain discriminative loss to fine-tune the embedding network to the target data. Similar to ADDA, we implement two models for PFAN (1) PFAN-LSTM and (2) PFAN-CNN.
- Deep neural network models without domain adaptation include (a) Plain LSTM (LSTM) (Hochreiter and Schmidhuber 1997), which is a traditional LSTM model that has not been pretrained by any data for domain adaptation. Similar to LSTM-W, LSTM has 100 units, 50% dropout, binary cross-entropy loss function, Adam optimiser and sigmoid activation. (b) Plain CNN (CNN) is similar to CNN-W, but it has not been pretrained by any data for domain adaptation. (c) Feedforward deep neural network (DNN) (Glorot and Bengio 2010). It has five hidden layers, each layer containing eighty units, 50% dropout applied to the input layer and the first two hidden layers, softmax activation and 0.04 learning rate. For all neural network-based models, hyperparameters are manually tuned based on cross-validation.
- Seven non-NN models including Support Vector Machines (SVM) (Hearst et al. 1998) (linear SVM (SVM-L) and nonlinear SVM (SVM-N)), Random Forest (RF) (Liaw and Wiener 2002), XGBoost (XGB) (Chen and Guestrin 2016), Multinomial Naive Bayes (MNB) (Lewis 1998), k-Nearest Neighbours (kNN) (Weinberger and Saul 2009) and Ridge Classifier (RC) (Hoerl and Kennard 1970). Hyperparameters of all these models are

[6] https://pytorch.org/.

[7] https://jupyter.org/.

[8] High-performance computing facilities used in this research were provided by eResearch Office, Queensland University of Technology, Brisbane, Australia

Table 1 Comparing models for domain adaptation

	TP	TN	FP	FN	Ac	Pr	Re	F_1	CK	AUC
LSTM-DA	3215	2152	1746	683	**0.688**	**0.648**	0.825	**0.726**	**0.377**	**0.688**
LSTM-W	2671	1968	1930	1227	0.595	0.581	0.685	0.629	0.190	0.595
LSTM	1793	2520	1378	2105	0.553	0.565	0.460	0.507	0.106	0.553
ADDA-LSTM	1021	2879	1019	2877	0.500	0.500	0.262	0.344	0.001	0.500
ADDA-CNN	**3809**	229	3669	**89**	0.518	0.509	**0.977**	0.670	0.036	0.518
PFAN-LSTM	3438	790	3108	460	0.542	0.525	0.882	0.658	0.085	0.542
PFAN-CNN	1143	**2916**	**982**	2755	0.521	0.538	0.293	0.380	0.041	0.521
CNN	2612	1994	1904	1286	0.591	0.578	0.670	0.621	0.182	0.591
CNN-W	2783	1890	2008	1115	0.599	0.581	0.714	0.641	0.199	0.599
DNN	2667	1771	2127	1231	0.569	0.556	0.684	0.614	0.139	0.569
XGB	2366	1873	2025	1532	0.544	0.539	0.607	0.571	0.087	0.544
RF	2329	2123	1775	1569	0.571	0.567	0.597	0.582	0.142	0.571
SVM-L	2291	2064	1834	1607	0.559	0.555	0.588	0.571	0.117	0.559
SVM-N	2539	1964	1934	1359	0.578	0.568	0.651	0.607	0.155	0.578
kNN	1596	2525	1373	2302	0.529	0.538	0.409	0.465	0.057	0.529
MNB	2851	1820	2078	1047	0.599	0.578	0.731	0.646	0.198	0.599
RC	1725	2432	1466	2173	0.533	0.541	0.443	0.487	0.066	0.533

automatically tuned using tenfold cross-validation and GridSearch using scikit-learn library.

None of the models, except LSTM-DA, ADDA, PFAN, LSTM-W and CNN-W, are pretrained or utilised any of the unlabelled datasets.

4.4 Effectiveness of domain adaptation

Table 1 compares 16 models with the proposed model LSTM-DA for domain adaptation. Best performing results are shown as bold in the table. LSTM-DA provides significantly better performance than all other models with improvements in AUC performance in the range of 14% to 27%. The second best results are obtained using MNB and CNN-W, and the next best results are obtained from LSTM-W. It is interesting to note that MNB performs equally well as simple domain adapted models, in comparison with the rest of the methods that had no domain adaptation. MNB is a generative model that learns underlying data distribution for making the prediction. Due to the capability to learn underlying data distributions, MNB shows a better generalisation for the unknown domain.

Both CNN-W and LSTM-W use pretrained word vectors (Word2Vec) for domain adaptation. As discussed in Sect. 3.1, Word2Vec is a simpler version of language model. Using pretrained word vectors was a popular transfer learning method (Bashar et al. 2018) before breakthrough for transfer learning occurred through the language model (Howard and Ruder 2018; Devlin et al. 2018). Therefore, it is not surprising that CNN-W and LSTM-W give reasonably good results. CNN is well known for learning varying length patterns similar to nGrams. On the other hand, LSTM is well known for learning long sequences. Because of varying length patterns, CNN generalise better than LSTM when the labelled training set is small (Bashar et al. 2018). Therefore, CNN-W performs better than LSTM-W and CNN perform better than LSTM.

Ensemble decision tree models such as RF and XGB are well known for their performance in classification tasks. However, experimental results in Table 1 show that they fail to generalise when the distribution of the target domain shifts from the source domain.

Experimental results in Chen et al. (2019), Tzeng et al. (2017) show that PFAN and ADDA work well in computer vision when there are enough labelled data in the source domain. However, results in Table 1 show that ADDA (ADDA-LSTM and ADDA-CNN) and PFAN (PFAN-LSTM and PFAN-CNN) do not work well when the source domain has limited information of labelled data. Results show that ADDA-CNN predicts most of the tweets as Positive that caused it adversely to account for the highest number of FP. Even though ADDA-CNN achieved the highest number of TP and Recall by predicting most tweets as positive, its Accuracy, Precision, Cappa Kohen and AUC are poor and F1 score is average, in comparison with LSTM-DA. On the other hand, PFAN-CNN predicts most of the tweets as negative that caused it adversely to account for the second highest number of FN. Even though PFAN-CNN achieved the highest number of TN by predicting most tweet as Negative, its Accuracy, Precision, Recall, F1, Cappa Kohen and AUC are poor in comparison with LSTM-DA. In summary, these methods produced highly skewed outcomes.

Table 2 Effect of Dataset Order in Progressive Domain Adaptation (W: Wiki103, R: RGT, C: CAST; W→R→C means $\mathcal{L}$ is first pre-trained with W, then R, then C)

	TP	TN	FP	FN	Ac	Pr	Re	F_1	CK	AUC
W	3172	1911	1987	726	0.652	0.615	0.814	0.700	0.304	0.652
W→R	**3409**	1714	2184	**489**	0.657	0.610	**0.875**	0.718	0.314	0.657
W→R→C	3215	**2152**	**1746**	683	**0.688**	**0.648**	0.825	**0.726**	**0.377**	**0.688**
W→C	3339	1867	2031	559	0.668	0.622	0.857	0.721	0.336	0.668
W→C→R	3055	1967	1931	843	0.644	0.613	0.784	0.688	0.288	0.644

Table 3 Comparing models for in-domain performance

	TP	TN	FP	FN	Ac	Pr	Re	F_1	CK	AUC
LSTM-DA	351	318	**60**	51	**0.858**	**0.854**	0.873	**0.863**	**0.715**	**0.857**
LSTM-W	346	297	103	34	0.824	0.771	0.911	0.835	0.650	0.827
LSTM	309	302	98	71	0.783	0.759	0.813	0.785	0.567	0.784
CNN-W	**356**	274	126	**24**	0.808	0.739	**0.937**	0.826	0.618	0.811
CNN	338	277	123	42	0.788	0.733	0.889	0.804	0.579	0.791
DNN	316	298	102	64	0.787	0.756	0.832	0.792	0.575	0.788
XGB	310	310	90	70	0.795	0.775	0.816	0.795	0.590	0.795
RF	309	316	84	71	0.801	0.786	0.813	0.799	0.603	0.802
SVM-L	281	307	93	99	0.754	0.751	0.739	0.745	0.507	0.753
SVM-N	298	**324**	76	82	0.797	0.797	0.784	0.790	0.594	0.797
kNN	204	321	79	176	0.673	0.721	0.537	0.615	0.342	0.670
MNB	324	271	129	56	0.763	0.715	0.853	0.778	0.528	0.765
RC	273	322	78	107	0.763	0.778	0.718	0.747	0.524	0.762

Reasonable good results of MNB, CNN-W and LSTM-W implies that generative modes have a better potential for generalisation to different domains. The proposed LSTM-DA model uses a sophisticated language model $\mathcal{L}$ that was progressively pretrained to learn domain invariant latent deep features. Therefore, when the class discriminative features are aligned with domain invariant features during the classification training, its performance in learning the decision boundary by maximising the likelihood is significantly better than any other models.

4.5 Ablation study of progressive domain adaptation

We conducted a set of experiments to validate the effect of different combinations of datasets used to pretrain the language model $\mathcal{L}$. The experimental results presented in Table 2 show that domain adaptation results are best for LSTM-DA when $\mathcal{L}$ is progressively pretrained with W→R→C (i.e. general → source domain-relevant specific → target domain-relevant specific). This combination produces improved performance in comparison with the combinations of W→C→R, W→C, W→R and W. A better performance with W→R→C than W confirms our conjecture that multiple relevant datasets can yield improved performance for domain adaptation than using a huge dataset only. The next best performance is obtained with W→C, which is better than W→R. It indicates that knowledge captured by $\mathcal{L}$ from general to target domain-relevant specific datasets is more useful for domain adaptation than the knowledge captured from general to source domain-relevant specific datasets. The poorest performance obtained with W→C→R indicates going from target-relevant specific to source domain-relevant specific datasets harms the performance. Best performing results are shown as bold in the table.

4.6 In-domain performance of models

For in-domain performance analysis, a pretrained model is fine-tuned, trained and tested (applied) with datasets that come from the same domain, i.e. these datasets have the same underlying data distribution. This analysis investigates how progressively domain adapted model compares with other models when trained and applied in the same domain. Experimental analysis in Sect. 4.4 shows that LSTM-DA, the proposed progressive domain adaptation model, performed significantly better than all other models. Well-known classification models such as XGBoost, RF, LSTM and CNN performed very poorly when the domain is changed. We want to test whether the poor performance of these well-known models resulted only because of domain shift or there are any other issues. In-domain performance analysis will highlight other hidden issues.

Table 4 Fear prediction results

TP	TN	FP	FN	Ac	Pr	Re	F1	CK	AUC
67	227	24	71	0.756	0.736	0.486	0.585	0.422	0.695

Table 5 Filtering keywords

List of fear-related keywords (Lambert et al. 2014)
afraid, fear, feared, fearful, fearing, fears, frantic, fright, horr, panic, scare, scaring, scary, terrified, terrifies, terrify, terror
List of East Asia-related keywords (Vidgen et al. (2020)
chinavirus, wuhan, wuhanvirus, chinavirusoutbreak, wuhancoronavirus, wuhaninfluenza, wuhansars, chinacoronavirus, wuhan2020, chinaflu, wuhanquarantine, chinesepneumonia, coronachina, wohan

For in-domain performance comparison, we use the target dataset D^t, EAHD. We use the same stratified dataset, split into 80/10/10 for training/validation/testing, created by data creators (Vidgen et al. 2020). Their empirical analysis revealed the best performance (F_1 of 0.83) with RoBERTa (BERT like transfer learning model) and the F_1 score of 0.76 with the baseline LSTM (Vidgen et al. 2020).

The results of our experiments are shown in Table 3. Best performing results are shown as bold in the table. The proposed LSTM-DA model obtains F_1 of 0.863 and outperforms the rest of the models used in this paper as well as models used in the research where this dataset was created (Vidgen et al. 2020). CNN-W, LSTM-W and CNN yield higher Recall value than our model LSTM-DA; however, LSTM-DA has higher values for Ac, Pr, F_1, CK and AUC. In other words, CNN-W, LSTM-W and CNN yield a lot of false positives which implies these models are biased to the positive class.

The higher recall of CNN might be caused by the ability of CNN to learn varying patterns that can generalise well. The higher recall of CNN-W and LSTM-W might be caused by the generalisation capacity of linear model Word2Vec. It will be interesting to see the performance when a CNN model is trained with progressively domain adapted $\mathcal{L}$ and combine some features from Word2Vec (e.g. creating an extra channel in the CNN model that can take Word2Vec word embedding). We will investigate this in our future work. In-domain performance analysis shows that well-known models perform reasonably well. This implies that the poor performance of these models as discussed in Sect. 4.4 comes from domain shift that they are unable to learn, whereas the proposed model LSTM-DA can capture necessary domain invariance and disentanglement for target domain by the proposed transfer learning with data reflecting general domain, source domain and target domain.

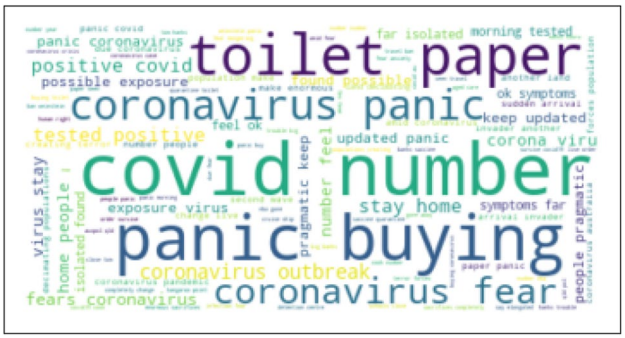

Fig. 7 Word cloud obtained from fear tweets in CAST Dataset

5 Case study: fear and hate spread during COVID-19

We conducted a case study by using the proposed LSTM-DA model to know how fear and hate spread during the COVID-19 pandemic in the Australian Twitter sphere. Is there a common pattern between fear and hate (i.e. East Asian hate)? We kept the same language model $\mathcal{L}$ tuned with datasets related to general, source and target domains and only changed D^s and D^t.

Fear Tweet Data We filter the 6.8 million tweets in the CAST dataset using 17 fear-related keywords, found effective in prior research (Lambert et al. 2014), as shown in Table 5. After filtering the CAST dataset with fear keywords, we are left with 147K (147,240) tweets. This data is the target dataset D^t in this case study.

Fear prediction with LSTM-DA model We used the domain adapted classification model $\mathcal{C}$ (using $\mathcal{L}$) to predict which of these CAST tweets are actual fear. We collected the labelled fear dataset from Mohammad and Kiritchenko (2018) as the source dataset D^s. The training set has 2252 (762 positive and 1490 negative) labelled tweets, and the validation set has 389 (138 positive and 251 negative) labelled tweets. The test set in the dataset is not labelled.

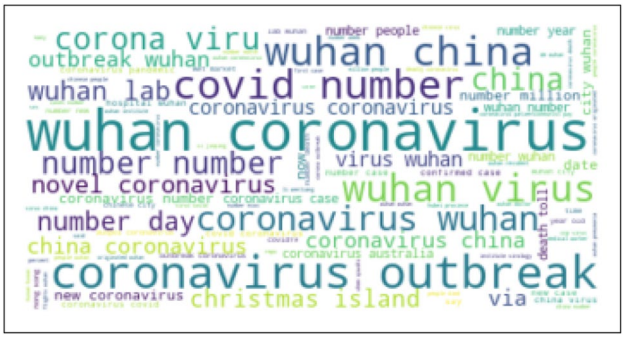

Fig. 8 Word cloud obtained from East Asian Hate Tweets in CAST

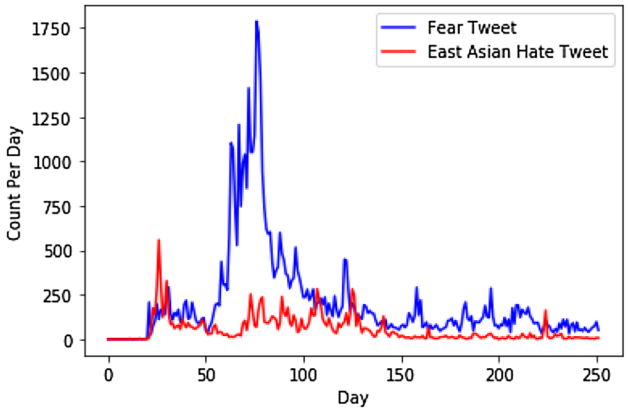

Fig. 9 Fear and hate distribution over the time

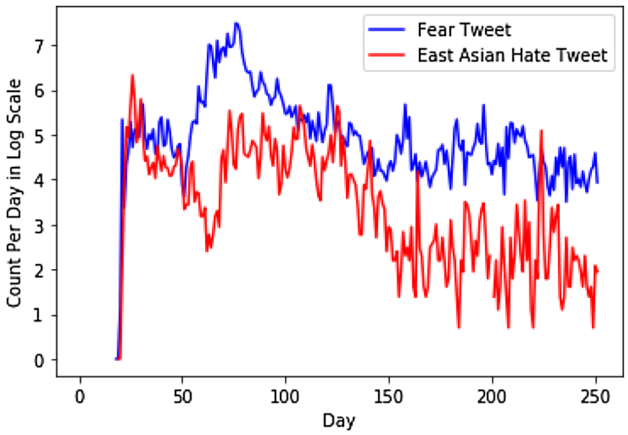

Fig. 10 Fear and hate distribution over the time in log scale

Therefore, we trained and validate $\mathcal{C}$ using the training set and test the model using the validation set. The experimental result on the validation set is provided in Table 4. We then applied this fear classifier to predict fear in D^t with 147K fear-related tweets.

The model predicted 53K (53156) tweets out of these 147K tweets as fear-relevant. Note, we have no labelled CAST data to validate the accuracy by standard measures. A word cloud generated from the fear tweets in CAST dataset is shown in Fig. 7. The distribution of these fear tweets over the period (i.e. 27th November 2019 to 7th September 2020) is shown in Fig. 9.

East Asia Hate Tweets Data We filter the 6.8 million tweets in the CAST dataset using 14 East Asia hate-related keywords, found effective in prior research (Vidgen et al. 2020), as shown in Table 5. Some of these keywords express anti-East Asian sentiments (e.g. *chinaflu*), and others (e.g. *wohan*) are neutral. After filtering the CAST dataset with East Asian hate keywords, we are left with 78K (78,508) tweets. This data is the target dataset D^t in this problem.

Hate prediction with LSTM-DA model We used the domain adapted classification model $\mathcal{C}$ (using $\mathcal{L}$) to predict which of these CAST tweets are actual East Asia hate. We used EAHD dataset (Vidgen et al. 2020) as the source dataset D^s.

The model predicted 13K (13920) tweets out of these 78K tweets as East Asia hate relevant. Note, we have no labelled CAST data to validate the accuracy by standard measures. Figure 8 shows a word cloud generated from the tweets predicted as East Asian hate by our model from the 13K tweets. Figure 9 shows how tweets predicted as East Asian hate are distributed over time. The distribution of these hate tweets over the period (i.e. 27th November 2019 to 7th September 2020) is shown in Fig. 9. In general, a **fear** peak is followed by an East Asian **hate** peak. For a closer observation, we have transformed this figure into log scale in Fig. 10.

Fear and Hate analysis Figures 9 and 10 show that counts of both fear and hate tweets fluctuate throughout the time, but both curves hit their peaks at the most active COVID-19 period in Australia (from February 2020 to May 2020) except the first peak of hate. The first peak of hate happened in January, fear was present during this time but not at its peak. This might be because at that time COVID-19 happened in China only and fear may have not emerged strongly in Australia.

We can observe from Figs. 9 and 10 that peak times of fear and hate during COVID-19 are generally similar. This can be explained as when people feel more fearful, they react in a negative way which may lead to hate or even anger. However, it can be seen that fear tweet count dominates hate tweet count over the COVID-19 period. This is strong evidence that fear is a primal human emotion when people face an uncertain situation that may put their lives under threat. These fears may relate to various things, job

loss, economic stagnation, isolation, along fears regarding their health and lives. More importantly, people may not always hate what they fear and that hate is just a manifestation of fear.

Based on this analysis, it is unclear whether fear of COVID-19 has led people to hate East Asian people. However, it is safe to conclude that these two emotions are strongly related during the COVID-19 pandemic based on the CAST data.

6 Conclusion

We propose a novel concept of unsupervised progressive domain adaptation of a language model through multiple text datasets when the target domain does not have labelled data. Such a language model learns a deep feature representation that can capture necessary domain invariance and disentanglement for target domain adaptation by bridging between general domain, source domain and target domain. The deep features learned by the domain adapted language model are then used to train a classifier using a small labelled dataset from a related source domain dataset. Finally, we apply the trained classifier to a target domain dataset where labelled data is unavailable. We showcase the proposed method by applying for hate speech and fear detection during the COVID-19 pandemic on a large Twitter dataset where the labelled information is unavailable. Though the proposed model is evaluated on the problem of hate and fear tweet detection, the method is applicable to any other situation where it is difficult to get a labelled dataset.

A series of experiments were conducted to investigate its effectiveness. When the classifier is trained on the domain adapted language model, the classifier performs significantly better than other state-of-the-art models. Theoretical analysis and experimental results show that the domain adaptation of the language model to learn necessary domain invariance and disentanglement for the target domain can significantly impact the accuracy of the classifier. Specifically, the domain adapted model LSTM-DA improved classification accuracy significantly when compared with the Word2Vec-based domain adaptation that captures features from the target domain.

By providing a Bayesian probability analysis of the proposed progressive domain adaptation, this paper implies the potential of domain adaptation through other models (e.g. CNN, Attention Models, etc.). Also, this implies the use of domain adaptation for selecting relevant features from the unlabelled datasets that can improve the semantics of available and missing features in a small labelled dataset. Bayesian probability analysis of the model can also be useful to identify and estimate uncertainties in the domain adaptation to choose the right datasets and models for pretraining, fine-tuning and training. This will be our future direction of research.

References

Al-garadi MA, Khan MS, Varathan KD, Mujtaba G, Al-Kabsi AM (2016) Using online social networks to track a pandemic: A systematic review. J Biomed Inform 62:1–11

Badjatiya P, Gupta S, Gupta M, Varma V (2017) Deep learning for hate speech detection in tweets

Badjatiya P, Gupta S, Gupta M, Varma V (2017) Deep learning for hate speech detection in tweets. In: Proceedings of the 26th International Conference on World Wide Web Companion, ser. WWW '17 Companion. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, pp 759–760

Baktashmotlagh M, Harandi MT, Lovell BC, Salzmann M (2013) Unsupervised domain adaptation by domain invariant projection, pp 769–776

Balasubramaniam T, Nayak R, Bashar MA (2020) Understanding the spatio-temporal topic dynamics of covid-19 using nonnegative tensor factorization: A case study. arXiv preprint arXiv:2009.09253

Banko M, Brill E (2001) Mitigating the paucity-of-data problem: Exploring the effect of training corpus size on classifier performance for natural language processing

Bashar MA, Nayak R (2021) Active learning for effectively fine-tuning transfer learning to downstream task. ACM Trans Intell Syst Technol (TIST) 12(2):1–24

Bashar MA, Nayak R, Suzor N (2020) Regularising lstm classifier by transfer learning for detecting misogynistic tweets with small training set. Know Inform Syst 62(10):4029–4054

Bashar MA, Nayak R, Balasubramaniam T (2020) Topic, sentiment and impact analysis: Covid19 information seeking on social media. arXiv preprint arXiv:2008.12435

Bashar MA, Nayak R, Suzor N, Weir B (2018) Misogynistic tweet detection: Modelling cnn with small datasets. In *Australasian Conference on Data Mining*. Springer, Berlin, pp 3–16

Bashar MA, Nayak R (2019) Qutnocturnal@ hasoc'19: Cnn for hate speech and offensive content identification in hindi language. In: CEUR Workshop Proceedings, vol 2517. CEUR-WS, pp 237–245

Blundell C, Cornebise J, Kavukcuoglu K, Wierstra D (2015) Weight uncertainty in neural network. pp 1613–1622

Bradbury J, Merity S, Xiong C, Socher R (2016) Quasi-recurrent neural networks

Brindha MD, Jayaseelan R, Kadeswara S (2020) Social media reigned by information or misinformation about covid-19: a phenomenological study

Chen C, Xie W, Huang W, Rong Y, Ding X, Huang Y, Xu T, Huang J (2019) Progressive feature alignment for unsupervised domain adaptation. pp 627–636

Chen T, Guestrin C (2016) Xgboost: A scalable tree boosting system. pp 785–794

Davidson T, Warmsley D, Macy M, Weber I (2017) Automated hate speech detection and the problem of offensive language

Devlin J, Chang M-W, Lee K, Toutanova K (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805

Founta AM, Chatzakou D, Kourtellis N, Blackburn J, Vakali A, Leontiadis I (2019). A unified deep learning architecture for abuse

detection. Association for Computing Machinery, New York, NY, USA, pp 105–114
Gal Y (2016) Uncertainty in deep learning. University of Cambridge, Cambridge
Gambäck B, Sikdar UK (2017) Using convolutional neural networks to classify hate-speech. In: Proceedings of the First Workshop on Abusive Language Online. Vancouver, BC, Canada: Association for Computational Linguistics, pp. 85–90
Ganin Y, Lempitsky V (2015) Unsupervised domain adaptation by backpropagation. In: International conference on machine learning. PMLR, pp. 1180–1189
Ghifary M, Kleijn WB, Zhang M (2014) Domain adaptive neural networks for object recognition. In: Pacific Rim international conference on artificial intelligence. Springer, Berlin, pp 898–904
Glorot X, Bordes A, Bengio Y (2011) Domain adaptation for large-scale sentiment classification: a deep learning approach. pp 513–520
Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. pp 249–256
Gong B, Grauman K, Sha F (2013) Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation, pp 222–230
Han X, Eisenstein J (2019) Unsupervised domain adaptation of contextualized embeddings for sequence labeling
Hausman DM, Woodward J (1999) Independence, invariance and the causal markov condition. Br J Philos Science 50(4):521–583
He R, Lee WS, Ng HT, Dahlmeier D (2018) Exploiting document knowledge for aspect-level sentiment classification arXiv preprint arXiv:1806.04346
Hearst MA, Dumais ST, Osuna E, Platt J, Scholkopf B (1998) Support vector machines. IEEE Intell Syst Appl 13(4):18–28
Heverin T, Zach L (2012) Law enforcement agency adoption and use of twitter as a crisis communication tool. In: Crisis Information Management. Elsevier, pp. 25–42
Higgins I, Amos D, Pfau D, Racaniere S, Matthey L, Rezende D, Lerchner A (2018) Towards a definition of disentangled representations. arXiv preprint arXiv:1812.02230
Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
Hoerl AE, Kennard RW (1970) Ridge regression: applications to non-orthogonal problems. Technometrics 12(1):69–82
Hoffman J, Tzeng E, Park T, Zhu J-Y, Isola P, Saenko K, Efros A, Darrell T (2018) Cycada: Cycle-consistent adversarial domain adaptation. In: International conference on machine learning. PMLR, pp. 1989–1998
Howard J, Ruder S (2018) Universal language model fine-tuning for text classification. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol 1, pp 328–339
Jozefowicz R, Vinyals O, Schuster M, Shazeer N, Wu Y (2016) Exploring the limits of language modeling. arXiv preprint arXiv:1602.02410
Kuncoro A, Dyer C, Hale J, Yogatama D, Clark S, Blunsom P (2018) Lstms can learn syntax-sensitive dependencies well, but modeling structure makes them better. pp 1426–1436
Lambert AJ, Eadeh FR, Peak SA, Scherer LD, Schott JP, Slochower JM (2014) Toward a greater understanding of the emotional dynamics of the mortality salience manipulation: Revisiting the "affect-free" claim of terror management research. J Person Soci Psychol 106(5):655
Lewis DD (1998) Naive (bayes) at forty: The independence assumption in information retrieval. In: European conference on machine learning. Springer, Berlin, pp. 4–15
Li Y, Gal Y (2017) Dropout inference in bayesian neural networks with alpha-divergences. In: Proceedings of the 34th International Conference on Machine Learning, vol 70. JMLR. org, pp. 2052–2061
Li Z, Wei Y, Zhang Y, Zhang X, Li X (2019) Exploiting coarse-to-fine task transfer for aspect-level sentiment classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 33, pp 4253–4260
Liaw A, Wiener M et al (2002) Classification and regression by randomforest. R news 2(3):18–22
Long M, Wang J, Ding G, Sun J, Yu PS (2013) Transfer feature learning with joint distribution adaptation, pp 2200–2207
Long M, Cao Y, Wang J, Jordan M (2015) Learning transferable features with deep adaptation networks. In: International conference on machine learning. PMLR, pp 97–105
MacAvaney S, Yao H-R, Yang E, Russell K, Goharian N, Frieder O (2019) Hate speech detection: Challenges and solutions. PloS One 14(8):e0221152
MacAvaney S, Yao H-R, Yang E, Russell K, Goharian N, Frieder O (2019) Hate speech detection: challenges and solutions. PLOS ONE 14(8):1–16
MacKay DJ (1992) A practical bayesian framework for backpropagation networks. Neural Comput 4(3):448–472
Malmasi S, Zampieri M (2017) Detecting hate speech in social media, CoRR, vol. abs/1712.06427. [Online]. Available: http://arxiv.org/abs/1712.06427
Melis G, Dyer C, Blunsom P (2017) On the state of the art of evaluation in neural language models. arXiv preprint arXiv:1707.05589
Merity S, Keskar NS, Socher R (2017) Regularizing and optimizing LSTM language models
Merity S, Xiong C, Bradbury J, Socher R (2016) Pointer sentinel mixture models
Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. pp 3111–3119
Mikolov T, Karafiát M, Burget L, Černockỳ J, Khudanpur S (2010) Recurrent neural network based language model
Mnih A, Yuecheng Z, Hinton G (2009) Improving a statistical language model through non-linear prediction. Neurocomputing 72(7–9):1414–1418
Mohammad S, Kiritchenko S (2018) Understanding emotions: A dataset of tweets to study interactions between affect categories. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)
Mozafari M, Farahbakhsh R, Crespi N (2020) A bert-based transfer learning approach for hate speech detection in online social media. In: Cherifi H, Gaito S, Mendes JF, Moro E, Rocha LM (eds) Complex networks and their applications VIII. Springer International Publishing, Cambridge, pp 928–940
Pan SJ, Tsang IW, Kwok JT, Yang Q (2010) Domain adaptation via transfer component analysis. IEEE Trans Neural Netw 22(2):199–210
Pan SJ, Yang Q (2009) A survey on transfer learning. IEEE Trans Know Data Eng 22(10):1345–1359
Park JH, Fung P (Aug. 2017) One-step and two-step classification for abusive language detection on Twitter. In: Proceedings of the First Workshop on Abusive Language Online. Vancouver, BC, Canada: Association for Computational Linguistics, pp. 41–45
Rajalakshmi R, Reddy B (2019) Dlrg@hasoc 2019: An enhanced ensemble classifier for hate and offensive content identification. In: FIRE
Rietzler A, Stabinger S, Opitz P, Engl S (2019) Adapt or get left behind: Domain adaptation through bert language model finetuning for aspect-target sentiment classification. arXiv preprint arXiv:1908.11860
Sharif Razavian A, Azizpour H, Sullivan J, Carlsson S (2014) Cnn features off-the-shelf: an astounding baseline for recognition. pp 806–813

Shimodaira H (2000) Improving predictive inference under covariate shift by weighting the log-likelihood function. J Stat Plan Infer 90(2):227–244

Tzeng E, Hoffman J, Saenko K, Darrell T (2017) Adversarial discriminative domain adaptation. pp 7167–7176

Vidgen B, Botelho A, Broniatowski D, Guest E, Hall M, Margetts H, Tromble R, Waseem Z, Hale S (2020) Detecting east asian prejudice on social media. arXiv preprint arXiv:2005.03909

Wang B, Wang A, Chen F, Wang Y, Kuo C-CJ (2019) Evaluating word embedding models: methods and experimental results. APSIPA Transactions on Signal and Information Processing, vol 8

Wang X, Schneider J (2014) Flexible transfer learning under support and model shift. pp 1898–1906

Waseem Z, Hovy D (2016) Hateful symbols or hateful people? predictive features for hate speech detection on twitter. pp 88–93

Waseem Z (2016) Are you a racist or am I seeing things? annotator influence on hate speech detection on Twitter. In: Proceedings of the First Workshop on NLP and Computational Social Science. Austin, Texas: Association for Computational Linguistics, pp 138–142

Weinberger KQ, Saul LK (2009) Distance metric learning for large margin nearest neighbor classification. J Mach Learn Res 10:207–244

Xu H, Liu B, Shu L, Yu PS (2019) Bert post-training for review reading comprehension and aspect-based sentiment analysis. arXiv preprint arXiv:1904.02232

Xu X, Zhou X, Venkatesan R, Swaminathan G, Majumder O (2019) d-sne: Domain adaptation using stochastic neighborhood embedding. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2497–2506

Yang Z, Chen W, Wang F, Xu B (2018) Unsupervised neural machine translation with weight sharing

Yosinski J, Clune J, Bengio Y, Lipson H (2014) How transferable are features in deep neural networks? In: Advances in neural information processing systems, pp 3320–3328