2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

A Transfer Learning Approach to Cross-Domain Author Profiling

Alexander G. Lain
University of Essex Online
Newbury, United Kingdom
al20884@essex.ac.uk

Ali M. S. Zalzala
Community Tracks Ltd
London, United Kingdom
zalzala@communitytracks.com

***Abstract*—Author profiling is the process of analysing text to determine one or more identifying characteristics of the author, mostly used to determine key demographic information. This type of classification problem is ideally suited to machine learning approaches. In this study, a new transfer learning approach is introduced using a pre-trained XLNet language model which was then fine-tuned to the specific author profiling task. Informed by previous literature, a Support Vector Machine, Feed-Forward Neural Network, and Convolution Neural Network were also developed for comparison. These algorithms were used to predict gender and age group on a single training and testing domain. As a model that works across multiple domains is desirable, each model was also tested on two domains which were independent of the training domain. The results demonstrated that the transfer learning model is superior to the other methods used for comparison in this study. Although applying the transfer learning model to the cross-domain context decreased its performance, it was still able to achieve a higher degree of accuracy on one testing domain than the Support Vector Machine which was trained and tested on that same domain. In addition, some interesting results emerged regarding the transfer of hyperparameter performance between tasks that share a common factor, be that classification task or training domain.**

Keywords—author profiling, machine learning, transfer learning, neural networks

I. Introduction

Author profiling (AP) is the process of determining various identifying characteristics of an author by analysing a piece, or pieces, of text which they have produced [1]. Developing an AP model involves identifying specific features from the text, building a standard representation for the target profile, and building a classification model [2]. This type of text classification problem is commonly approached using machine learning (ML) algorithms. The identified characteristics can include age, gender, native language, and political leaning. The applications of AP are wide ranging and include targeted advertising campaigns and suspect profiling [3].

One of the main challenges in ML driven AP is developing models which produce accurate results across multiple domains [4]. In particular, a model which could be trained on a single data-rich domain and then used to perform AP on multiple other domains would be highly desirable. This type of model would have the potential to be employed in a wide variety of contexts.

Several ML algorithms are used for AP. Earlier research [2] compared the effectiveness of two different ML approaches to author profiling in a social media context. The first approach used Support Vector Machine (SVM) algorithms and Feed-Forward Neural Networks (FFNN), and the second used Long Short-Term Memory (LSTM) algorithms and Convolutional Neural Networks (CNN). Their results show that the first approach, which used manually extracted data, was more effective than when learning from raw data.

Similarly, it was found that although Deep Learning (DL) approaches showed promise, the results were sub-par for the task of predicting well-being markers [6, 7]. A recent study [8] used modern techniques such as DL and Transfer Learning (TL) methods to perform mental illness classification on text from social media. Comparing this to earlier studies demonstrates the rapid improvement in DL techniques, particularly in a healthcare context.

The aim of this project is to produce a model which can successfully identify the age and gender of an author in two domains, namely the training domain and one other distinct domain. This will be achieved by (i) selecting the appropriate training dataset, (ii) selecting an ML algorithm, (iii) developing a functional model, and (iv) testing and analysing the model. This project will study word2vec and other commonly used word embedding models, as well as DL neural networks [29] such as Recurrent Neural Networks (RNN), LSTMs, and CNNs [30]. In addition, the study will investigate Transformers [31] [32], Transfer Networks [33], and Adaptive Networks [34]. Previous relevant theses work will provide useful guidance on the structure [5, 35, 36].

This project investigates the following questions as objectives:

- Can a ML AP model effectively predict an individual's age and gender?
- Can a DL neural network produce results comparable to those of a traditional ML approach?
- Can a TL approach be applied to such a DL neural network to further increase its performance?
- Can a model trained on one domain yield accurate results when used across multiple other domains?

II. Transfer Learning

TL is inspired by the way humans learn and aims to apply knowledge gained from one problem to solve a second, related problem [10]. These problems include tasks involving different languages or domains [11]. For example, a model trained to distinguish between images of birds and planes in flight could then be applied to the similar task of distinguishing between different species of birds in flight. This reduces the time and resources used to train models and provides a possible solution when the available training data for a given task is limited [12]. The literature review identified three main methods through which TL is currently implemented in AP, each with different strengths and limitations.

A. Universal Language Model Fine-Tuning (ULMFiT)

ULMFiT was originally proposed as an effective TL method which could be applied to any natural language processing (NLP) task [13]. The authors found their ULMFiT method reduced error rates by up to 24% when compared to other state-of-the-art methods at the time. In particular, the authors highlighted the potential effectiveness of their approach for tasks where labeled training data was limited.

Since its initial proposal, ULMFiT has been shown to be effective across a number of NLP tasks such as fake news spreader profiling [14], and less traditional tasks such as predicting yields from chemical reactions [15]. In addition to its wide usage, ULMFiT is often cited as a promising avenue for future investigation when other TL methods have been used in the research [16].

The Attentional Universal Language Model Fine-Tuning (AULMFiT) was proposed as an improvement over ULMFiT [16]. In this new model, the authors made changes to the classifier fine-tuning process. In particular, the average and max pooling operation is replaced by a soft attention layer. This allowed the model to identify more accurately important information which is not always average or maximum values. In their testing on six separate datasets, the new AULMFiT model outperformed the original ULMFiT across all tasks.

B. Bidirectional Encoder Representation from Transformers (BERT)

BERT was originally proposed as a new language representation model designed to pre-train deep bidirectional representations from unlabelled text [17]. This then allows the pre-trained BERT model to be fine-tuned with the addition of a single output layer. This method produced superior results when tested on a variety of NLP tasks.

Since its inception, BERT has become a popular choice when approaching NLP tasks [18]. Despite its popularity, the model is criticised for its inability to handle longer documents as well as for being computationally intensive which limits usability in some cases [19].

A number of BERT based models have since been produced which each improve on the original in certain ways. The DistilBERT (a distilled version of BERT) retains 97% of the language capabilities of the original, whilst being 40% smaller and 60% faster than the original [20]. The BERT-AAD (Adversarial Adaptation with Distillation) focused on cross-domain tasks [21] and has been shown to be effective in other works [22]. Finally, RoBERTa (Robustly Optimised BERT Approach) [23] outperformed the original BERT model when first proposed, and continues to perform highly across a range of NLP tasks when compared to other available models [24].

C. XLNet

XLNet [25] uses an autoregressive approach in order to combat issues faced in BERT due to ignoring dependency between masked positions. Testing the XLNet found it to outperform BERT in twenty tasks across a variety of NLP problems. XLNet has also been shown to provide similar levels of accuracy to RoBERTa when using the same hyperparameters [26]. In addition, it was reported that a XLNet-based model [27] outperformed a BERT-based model even without performing fine-tuning on domain specific data. This is particularly advantageous as it reduces the training time and complexity of the model, both of which are often used as arguments against the use of larger architectures such as XLNet [28].

D. Comparisons

The main limitations remain in the handling of long form text documents and the computationally intensive nature of training the models. These issues are particularly prevalent in the BERT-based models. This is in part due to the fine-tuning needed for these models. Despite these issues, BERT-based models have been used extensively in the literature and have consistently produced high degrees of accuracy.

XLNet is the most recently published method and has therefore not been used as extensively in other studies as the ULMFiT and BERT-based methods. Studies that have employed XLNet have found it to produce similar levels of accuracy to the RoBERTa method, along with comparable results without fine-tuning on domain specific data. This makes it a promising choice for use in this study where cross-domain performance is of interest.

III. Research Methodologies

A. Description of Datasets

This study has been carried out using secondary data which is used widely in the reviewed literature in order to facilitate comparisons with other research. The datasets were provided by PAN, a series of scientific events and shared tasks on digital text forensics and stylometry [3]. Each dataset is described as follows.

1) The PAN 2014 AP task data set. This data set is made up of four different parts: social media, Twitter, blogs, and hotel reviews [37]. These parts are referred to as PAN14-social, PAN14-twitter, PAN14-blogs, and PAN14-reviews throughout this paper.

2) The PAN 2015 AP task data set. This data set is made up of Twitter data [9]. This dataset is referred to as PAN15-twitter throughout this paper.

3) The PAN 2017 AP task data set. This data set is made up of Twitter data [38]. This dataset is referred to as PAN17-twitter throughout this paper.

B. Text Pre-Processing

Text pre-processing is the process of normalising text data before it is used in further analysis. This is particularly important when looking at data collected from social media sites, where users frequently ignore grammar rules and spelling, and use abbreviations, slang, and emoji [39]. Although it is agreed that text pre-processing is a crucial step in an AP pipeline that can greatly affect the performance of a model, there is no consensus on a single approach as this area is less explored in the literature when compared with feature extraction and classification [40].

It was found that the removal of URLs and stop words only minimally affected a classifier's performance [41], whereas lowercasing and lemmatisation were the best performing techniques when only a single technique was used [40]. It was also found that the order of the pre-processing components can significantly affect the performance of a classifier [39] [40, 42] . Additionally, both [40, 43] proposed a similar order in which to use the various pre-processing techniques which informs the how the text pre-processing will be completed for this study. The same pre-processing steps were used on all datasets and implemented using the spaCy NLP library. These steps are as follows: removal of HTML-tags, removal of URLs, user-mentions, and hashtag symbols, removing punctuation, lowercasing of words, removing numbers, removing stop words, and lemmatisation. Language detection was also used in order to remove entries in the dataset which were not thought to be English.

C. Feature Extraction

Feature extraction is the process of building a set of values (features) from an initial data set. In this case, the pre-processed text forms the initial dataset. The choice of feature extraction method varied based on which ML algorithm was used. This choice was discussed for each learning algorithm [5].

a) Term Frequency-Inverse Document Frequency (TF-IDF): TF-IDF is a statistical method that is used to identify the importance of any word in a single document of a corpus.

b) Word2vec: Word2vec is a word embedding technique that performs training using a two-layer neural network [44, 45].

D. Transfer Learning

TL repurposes a previously trained model as a starting point for the training a second model. This can be achieved by either fine-tuning or fixed feature extraction. Both methods will be experimented with during this study. Fine-tuning allows for all the weights in the network to be updated during training. Fixed feature extraction, however, only allows the weights on the final fully connected output layer to be updated, freezing the original model. As such, fine-tuning can allow for better generalisation to the second task at the cost of training time, whereas fixed feature extraction allows for faster training but may not achieve optimal results.

E. Benchmark Algorithms

There are numerous AP studies which have been conducted using the same PAN datasets used in this project. It is therefore informative to use the results from such studies as benchmarks to compare against, as listed in Table I.

The Lundqvist and Svensson [5] research and testing was conducted using the same PAN datasets used in this paper, along with the addition of a dataset consisting of Google blogs data. The linear SVM and CNN models were the highest performing ML and DL models respectively. Independently, From [46] used the PAN15-twitter dataset, augmented with the PANDORA dataset [47], in their study of BERT-based transformers. This mixed dataset produced a high level of accuracy in both classification tasks.

IV. Machine Learning Algorithms

A. Support Vector Machines

In this research the SVM model was implemented using the scikit-learn Python library [48]. Experiments were done using both the sklearn.svm.SVC and sklearn.svm.LinearSVC models. The LinearSVC model uses a linear kernel only and has a different implementation to SVC, which allows it to scale to larger datasets.

Experimentation was done on the C and γ values, and the RBF and linear kernels which can be passed into the SVC. Using the PAN15-twitter dataset to fine-tune the hyperparameters, it was found that $1 < C < 2$ and $\gamma = 1$ with the RBF kernel produced the most accurate models. The TF-IDF feature extraction was performed using the TfidVectorizer from scikit-learn.

TABLE I. Benchmark Algorithms from Studies and Their Highest Accuracy Results for Each Classification Task

Study	Learning Algorithm	Gender Classification Results	Age Group Classification Results
Lundeqvist & Svensson (2017)	Linear SVM	0.8333	0.4028
Lundeqvist & Svensson (2017)	CNN	0.7654	0.3909
From (2022)	BERT-based	0.915	0.891

B. Artificial Neural Networks (ANN)

The ANNs used in this research were implemented using the Sequential model available from the Tensorflow library [49]. The FFNN implementation experimented with both the number of hidden layers and the number of neurons in each layer. A network with two hidden layers containing 100 neurons was found to be optimal for the purposes of this research. All hidden layers use the ReLU activation function. The CNN used in this research used a single convolution layer, a max pooling layer, and a fully connected layer.

Testing was performed using both available Word2vec models which were pre-trained on independent datasets, and Word2vec models trained on the main training dataset. The Gensim library [50] was used to implement the Word2vec models. In the context of this work, it was found that the Word2vec models were not as effective as the more basic approach using the Keras Tokenizer. This tokeniser approach simply assigns an integer value to each unique word in a dataset.

C. Transfer Learning using XLNet

The XLNet model is far more computationally intensive than the ANNs and SVM outlined above. Therefore Google Colaboratory (Colab) was used for developing and testing this model. Colab allows access to specialised hardware including Google's own tensor processing units (TPUs) in order to significantly reduce training times.

The XLNet model used in this research was implemented using the pre-trained TFXLNetModel and XLNetTokenizer available in the transformers library from Hugging Face [51] and Tensorflow [49]. In particular, the pre-trained 'xlnet-large-cased' and 'xlnet-base-cased 'models were both tested. The "xlnet-base-cased" model was ultimately chosen for this work due to it being more lightweight. The learning rate and input length were also experimented with on the PAN15-twitter dataset.

The XLNet model allows sequences in the inputs to be separated as follows: A <sep> B <sep> <cls>. This can be used to either treat the entire input as a single sequence, or split the input sentences into individual sequences. In this research each input was treated as a single sequence.

V. Models and Results

Each model was first trained, validated, and tested on a single dataset (domain). Cross-domain performance was then tested using the PAN15-twitter dataset as the training domain, while testing was performed using the remaining datasets. As discussed in section I, being able to train a model on a data-rich domain which can be used for a task on a second data-poor domain would be highly beneficial. Twitter is one such data-rich domain, hence its use in these experiments.

In the case of the SVM, the dataset being used is split into a training set and a test set. In this research, a stratified k-fold cross validation ($k = 5$) approach was used to evaluate the performance of the SVM. While developing the FFNN, CNN, and XLNet models used in this research, each dataset used is split into three subsets using stratified random sampling. These are the training, validation, and test sets. The entire dataset is initially split into two using a 4:1 ratio. The larger subset is then again divided using the same ratio. The result of this is training (64%), validation (16%), and test (20%). When experimenting with cross-domain performance, training and validation is performed using one dataset, and testing is done using the other dataset. Therefore, the training and validation dataset can be split simply as training (80%) and validation (20%). During training, only the training set is used to adjust the weights in the network. The validation set is used to measure the performance of the model after each training epoch. The test set is used to evaluate the final model's performance. Evaluating a model's performance computes accuracy, precision, recall, and F_1-score using a confusion matrix.

A. Results of Gender Classification

The results displayed in Figure 1 show that XLNet produced the highest accuracy across all the gender classification tasks. The FFNN, CNN, and XLNet performed best when tested on the PAN14-blogs dataset in both the single and cross-domain task. The SVM performed best on the PAN14-reviews dataset in the single domain task, and the PAN14-blogs dataset in the cross-domain task. The SVM produced the most consistent performance across all gender classification tasks, when compared to the other algorithms.

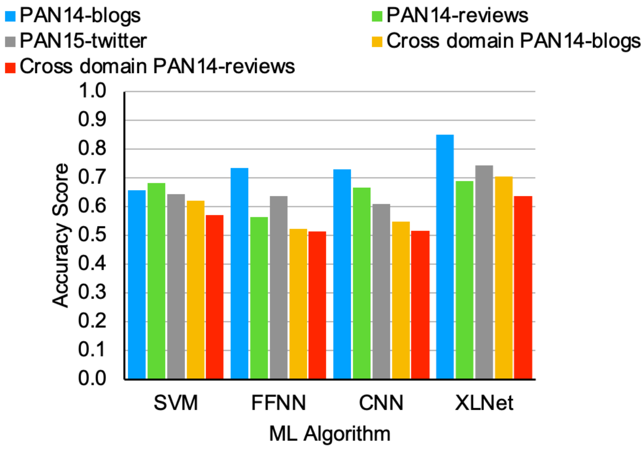

Fig. 1. A bar chart displaying the accuracy results of each ML algorithm across all single and cross domain gender classification tasks.

B. Results of Age Group Classification

The results displayed in Figure 2 show XLNet produced the highest accuracy results across all the age group classification tasks.

The experiment using PAN15-twitter produced highest accuracy scores, with each algorithm achieving over 0.5. The cross domain PAN14-blogs experiment produced the lowest scores across all four algorithms. In the case of the PAN14-reviews dataset, the FFN, CNN, and XLNet all produced higher accuracy when trained on PAN15-twitter first. No algorithm produced consistent results across all age group tasks.

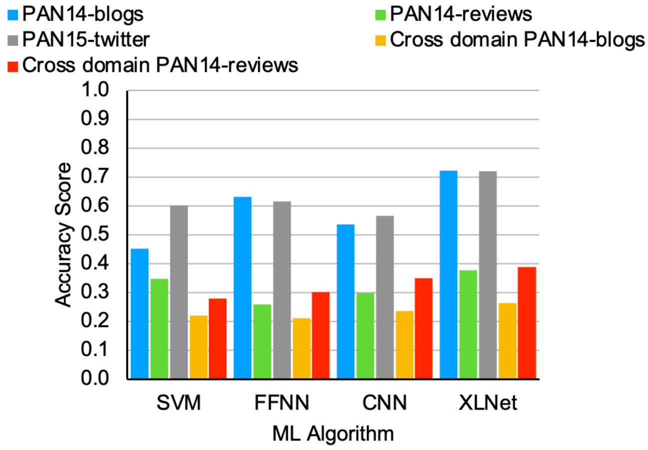

Fig. 2. A bar chart displaying the accuracy results of each ML algorithm across all single and cross domain age group classification tasks.

VI. Discussions

A. Data Pre-Processing

The performance of the SVM and FFNN were particularly affected by different pre-processing techniques. When applied to the two largest datasets used in this study, PAN14-social and PAN17-twitter, the execution time of the pre-processing pipeline became limiting. As the text pre-processing was not the focus of this study, once it enabled satisfactory initial results from the SVM and FFNN models then the focus shifted to hyperparameter tuning. This meant that the pre-processing pipeline was never optimised to reduce its time complexity. Taking advantage of the powerful vectorisation features available in Pandas [52] would enable this necessary reduction. This would allow for more iterative improvements to be made to this pipeline without execution time being a limiting factor.

The lack of such pre-processing techniques needing to be used when fine-tuning a pre-trained XLNet model makes it particularly attractive. In this research, the only text pre-processing applied to the inputs for an XLNet model was the removal of HTML tags. Additional tags can be added to the input of an XLNet model to break each input into individual sequences. Usually, each sentence in an input would be tagged as an individual sequence. XLNet is contextually aware across sequence boundaries, with the influence of a previous sequences fading as distance from it increases. It follows that a deeper understanding into a given input could be gleaned providing this sequence tagging is performed correctly.

B. Hyperparameters

The main issue with hyperparameter tuning [5, 46] is the cost, both in time and computing resources. Each algorithm has several adjustable hyperparameters. To reduce complexity, the hyperparameters for each algorithm were tuned while performing the gender classification task using the PAN15-twitter dataset. Finding the optimal values for any ML algorithm given a particular dataset and classification task would require a much more fine-grained approach than was feasible for this study.

The XLNet experiments for this research were conducted using Google's Colab platform in order to access the necessary TPUs for acceleration. This made large amounts of experimenting with hyperparameters infeasible as the resource limits would regularly be hit. Any further research into the effects of fine-tuning hyperparameters for pre-trained XLNet models should secure continuous access to the necessary GPUs or TPUs in order to avoid this issue.

C. Comparisons with Benchmark Algorithms

As illustrated in Figure 3, the cross-domain results for gender classification were lower than the corresponding single domain results for all models, which is to be expected. The XLNet model was again the top performer in both tests. The XLNet model trained on PAN15-twitter and tested on PAN14-blogs outperformed the SVM model which was both trained and tested on PAN14-blogs.

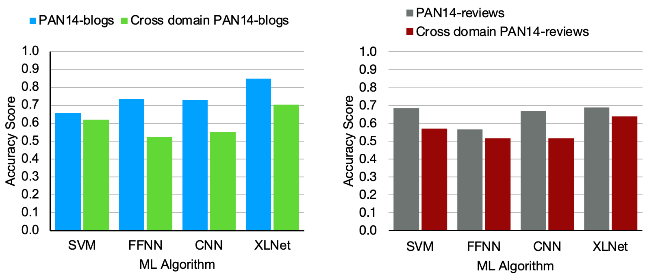

Fig. 3. Two bar charts displaying the single domain and cross domain accuracy scores for PAN14-blogs (left) and PAN14-reviews (right) in the gender classification task.

The cross-domain test results for age group classification were less satisfying. In the case of training on PAN15-twitter and testing on PAN14-blogs, the accuracies were lower for all models. As illustrated in Figure 4, the XLNet results when tested on PAN14-reviews were comparable for both the single and cross domain case.

These results can now be compared against the benchmark algorithms in section IV.E. Both the SVM and CNN reported [5] achieved higher accuracy scores than the SVM and CNN produced in this study. This is to be expected as their approach focused on optimising hyperparameters for each specific task and dataset. The difference in the accuracy of the SVMs was ~0.18, whereas for the CNNs it was only ~0.04. This suggests that an SVM may be more reliant on hyperparameter fine-tuning for top performance in binary classification tasks.

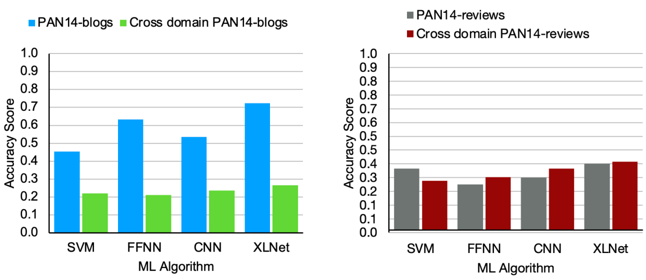

Fig. 4. Two bar charts displaying the single domain and cross domain accuracy scores for PAN14-blogs (left) and PAN14-reviews (right) in the age group classification task.

Both [5] and this study used the PAN14-blogs dataset for the age group classification task. The SVM and CNN used in this study both achieved higher accuracy scores than those reported [5]. This study achieved accuracy scores ~0.05 and ~0.14 higher for the SVM and CNN respectively. Both the reported SVM and CNN [5] were outperformed by the XLNet model for both classification tasks.

The reported BERT-based model [46] out-performed the XLNet model used in this research in both gender and age group classification. This demonstrates the import role that the datasets play in training and testing. No specific manual curation of the datasets was made before they were used in this study. In contrast, [46] removed certain data from the PAN15-twitter dataset before augmenting it with additional data selected from the PANDORA dataset [47].

VII. Conclusions

This study aimed to investigate the evolution of ML algorithms in AP. The study also aimed to investigate the application of new TL approaches to this context. Age group and gender classification were the tasks focused on. Additionally, the study investigated each model's performance for classification tasks on a domain which was distinct from its training domain. To achieve these four different ML algorithms were compared using a variety of datasets. Out of the four models which were evaluated, the TL XLNet model outperformed the SVM, FFNN, and CNN models in all ten experiments. Based on this, a fine-tuned XLNet model is the most effective author profiling method presented in this study.

In general, model performance is contingent on several factors including the balance and size of the dataset, text pre-processing, and hyperparameter selection. In the case of an XLNet model however, the reliance on text preprocessing falls away and satisfactory performance is achieved with little hyperparameter tuning. The presented results also show promise in the production of a cross domain classification model. This is however heavily reliant on the selection of an appropriate training domain with a suitable amount of available data and influence.

This research was limited in scope to gender and age group classification AP tasks only. The effectiveness of a pre-trained XLNet model when applied to other AP tasks should be investigated. These tasks could include more difficult multi class problems such as predicting native language, subjective wellbeing, and Big Five personality traits.

TL was the focus of this study, rather than the effects of the pretrained model's size on the results. The size of the XLNet model must play some role in the improved results, but it can be demonstrated that size is not the only indicator of performance. The nature of the model also plays a significant role. Using a number of the larger pre-trained Word2vec models showed no significant improvement when used with the SVM and FFNN over simply training our own Word2vec model with the datasets used in this work.

In this research the pre-trained XLNet model fed into a single fully connected output layer only, and the effects of different structures should be investigated. Additionally, task and dataset specific hyperparameter tuning should also be investigated as should the cross-domain potential of a fine-tuned XLNet model. Augmenting the PAN datasets to increase model performance [46] can investigate whether such mixed training data helps or hinders cross domain performance.

REFERENCES

1. M. Weigmann, Stein, B., & Potthast, M. , "Overview of the Celebrity Profiling Task at PAN," 2020. [Online]. Available: http://ceur-ws.org/Vol-2696/paper_259.pdf.
2. A. López-Monroy, MOntes-y-Gómez, M., Escalante, H., Villaseñor-Pineda, L., & Stamatatos, E., "Discriminative subprofile-specific representations for author profiling in social media.," *Knowledge-Based Systems,* vol. 89, pp. 134-147, 2015, doi: https://doi.org/10.1016/j.knosys.2015.06.024.
3. PAN, "PAN," N.D. [Online]. Available: https://pan.webis.de/.
4. J. P. Neto, I., "Multi-source BERT stack ensemble for cross-domain author profiling.," 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/exsy.12869.
5. E. Lundeqvist and M. Svensson, "Author profiling: A machinelearning approach towards detectinggender, age and native languageof users in social media (Dissertation)," 2017.
6. C. Yang and P. Srinivasan, "Life Satisfaction and the Pursuit of Happiness on Twitter," *PLOS ONE,* vol. 11, no. 3, p. e0150881, 2016, doi: 10.1371/journal.pone.0150881.
7. L. Chen, T. Gong, M. Kosinski, D. Stillwell, and R. L. Davidson, "Building a profile of subjective well-being for social media users," *PLOS ONE,* vol. 12, no. 11, p. e0187278, 2017, doi: 10.1371/journal.pone.0187278.
8. M. A. Iqra Ameer, Grigori Sidorov, Helena Go ́mez-Adorno, Alexander Gelbukh, "Mental Illness Classification on Social Media Texts using Deep Learning and Transfer Learning," 2022, doi: 10.48550/arxiv.2207.01012.
9. F. e. a. Rangel, "Overview of the 3rd Author Profiling Task at PAN 2015.," 2015. [Online]. https://pan.webis.de/downloads/publications/papers/rangel_2015.pdf.
10. K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big Data,* vol. 3, no. 1, 2016, doi: 10.1186/s40537-016-0043-6.
11. P. Azunre, Transfer Learning for Natural Language Processing. Manning, 2021.
12. D. S. Maitra, U. Bhattacharya, and S. K. Parui, "CNN based common approach to handwritten character recognition of multiple scripts," 2015: IEEE, doi: 10.1109/icdar.2015.7333916. [Online]. Available: https://dx.doi.org/10.1109/icdar.2015.7333916
13. J. R. Howard, S., "Universal Language Model Fine-tuning for Text Classification," 2018, doi: 10.48550/arxiv.1801.06146.
14. H. L. Shashirekha, & Baluchzhai, F., "ULMFiT for Twitter Fake News Spreader Profiling Notebook for PAN at CLEF 2020.," 2020. [Online]. Available: https://www.researchgate.net/publication/359024571_ULMFiT_for_Twitter_Fake_News_Spreader_Profiling_Notebook_for_PAN_at_CLEF_2020.
15. S. Singh and R. B. Sunoj, "A transfer learning protocol for chemical catalysis using a recurrent neural network adapted from natural language processing," *Digital Discovery,* vol. 1, no. 3, pp. 303-312, 2022, doi: 10.1039/d1dd00052g.
16. A. F. Sotelo, Gómez-Adorno, H., Esquivel-Flores, O., Bel-Enguix, G., "Gender Identification in Social Media Using Transfer Learning.," K. Figueroa Mora, Anzurez Marín, J., Cerda, J., Carrasco-Ochoa, J., Martínez-Trinidad, J., Olvera-López, J. Ed., 2020.
17. M.-W. C. Jacob Devlin, Kenton Lee, Kristina Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," 2019, doi: 10.48550/arxiv.1810.04805.
18. B. B. Wentao Yu and a. D. Kolossa, "BERT-based ironic authors profiling," 2022.
19. M. Polignano, M. De Gemmis, and G. Semeraro, "Contextualized BERT Sentence Embeddings for Author Profiling: The Cost of Performances," Springer International Publishing, 2020, pp. 135-149.
20. L. D. Victor SANH, Julien CHAUMOND, Thomas WOLF, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," 2020, doi: 10.48550/arxiv.1910.01108.
21. K. L. Minho Ryu, "Knowledge Distillation for BERT Unsupervised Domain Adaptation," 2020, doi: 10.48550/arxiv.2010.11478.
22. I. P. José Pereira Delmondes Neto, "Multi-source BERT stack ensemble for cross-domain authorprofiling," 2021.
23. M. O. Yinhan Liu, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," 2019, doi: 10.48550/arxiv.1907.11692.
24. P. Rajapaksha, R. Farahbakhsh, and N. Crespi, "BERT, XLNet or RoBERTa: The Best Transfer Learning Model to Detect Clickbaits," *IEEE Access,* vol. 9, pp. 154704-154716, 2021, doi: 10.1109/access.2021.3128742.
25. Z. D. Zhilin Yang, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, Quoc V. Le, "XLNet: Generalized Autoregressive Pretraining for Language Understanding," 2020. [Online]. Available: https://arxiv.org/pdf/1906.08237.pdf.
26. A. F. Adoma, N.-M. Henry, and W. Chen, "Comparative Analyses of Bert, Roberta, Distilbert, and Xlnet for Text-Based Emotion Recognition," 2020: IEEE, doi: 10.1109/iccwamtip51612.2020.9317379. [Online]. Available: https://dx.doi.org/10.1109/iccwamtip51612.2020.9317379.
27. M. Mustapha, K. Krasnashchok, A. Al Bassit, and S. Skhiri, "Privacy Policy Classification with XLNet (Short Paper)," Springer International Publishing, 2020, pp. 250-257.
28. P. R. Aditya Malte, "Evolution of Transfer Learning in Natural Language Processing," 2019, doi: 10.48550/arxiv.1910.07370.
29. Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature,* vol. 521, no. 7553, pp. 436-444, 2015, doi: 10.1038/nature14539.
30. D. Goularas and S. Kamis, "Evaluation of Deep Learning Techniques in Sentiment Analysis from Twitter Data," presented at the 2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML), 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8876896/.
31. K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, "Transformer in transformer," *Advances in Neural Information Processing Systems,* vol. 34, pp. 15908-15919, 2021.
32. T. Wolf *et al.*, "HuggingFace's Transformers: State-of-the-art Natural Language Processing," p. arXiv:1910.03771. [Online]. Available: https://ui.adsabs.harvard.edu/abs/2019arXiv191003771W.
33. M. Qiu *et al.*, "Easytransfer: A simple and scalable deep transfer learning platform for NLP applications," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 4075-4084.
34. C. Cortes, X. Gonzalvo, V. Kuznetsov, M. Mohri, and S. Yang, "Adanet: Adaptive structural learning of artificial neural networks," in *International conference on machine learning*, 2017: PMLR, pp. 874-883.
35. L. Kaati, E. Lundeqvist, A. Shrestha, and M. Svensson, "Author Profiling in the Wild," in *2017 European Intelligence and Security Informatics Conference (EISIC)*, 11-13 Sept. 2017 2017, pp. 155-158, doi: 10.1109/EISIC.2017.32.
36. S. Novinfard, "Profiler Application using Sentiment Analysis," in "MSc Computer Science - Software Engineering with Industrial Experience " Queen Mary, University of London, 2018.
37. F. e. a. Rangel, "Overview of the 2nd Author Profiling Task at PAN 2014," 2014. [Online]. Available: https://pan.webis.de/downloads/publications/papers/rangel_2014.pdf.
38. F. e. a. Rangel, "Overview of the 5th Author Profiling Task at PAN 2017: Gender and Language Variety Identification in Twitter.," 2017. [Online]. Available: https://pan.webis.de/downloads/publications/papers/rangel_2017.pdf.
39. M. A. Palomino and F. Aider, "Evaluating the Effectiveness of Text Pre-Processing in Sentiment Analysis," *Applied Sciences,* vol. 12, no. 17, p. 8765, 2022, doi: 10.3390/app12178765.
40. U. Naseem, Razzak, I. & Eklund, P.W., "A survey of pre-processing techniques to improve short-text quality: a case study on hate speech detection on twitter.," *Multimed Tools Appl,* vol. 80, pp. 35239–35266, 2021, doi: https://doi.org/10.1007/s11042-020-10082-6.
41. Z. Jianqiang and G. Xiaolin, "Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis," *IEEE Access,* vol. 5, pp. 2870-2879, 2017, doi: 10.1109/access.2017.2672677.
42. U. H. Hair Zaki, R. Ibrahim, S. Abd Halim, and I. I. Kamsani, "Text Detergent: The Systematic Combination of Text Pre-processing Techniques for Social Media Sentiment Analysis," Springer International Publishing, 2022, pp. 50-61.
43. C. Zhai and S. Massung, Text Data Management and Analysis: A Practical Introduction to Information Retrieval and Text Mining (ACM Books). San Rafael: Morgan & Claypool Publishers, 2016.
44. K. C. Tomas Mikolov, Greg Corrado, Jeffrey Dean, "Efficient Estimation of Word Representations in Vector Space," 2013, doi: 10.48550/arxiv.1301.3781.
45. K. C. Tomas Mikolov. Ilya Sutskever, Greg Corrado, Jeffrey Dean, "Distributed Representations of Words and Phrases and their Compositionality," 2013, doi: 10.48550/arxiv.1310.4546.
46. V. From, "Transfer Learning for Automatic Author Profiling with BERT Transformers and GloVe Embeddings.," 2022. [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1645358/FULLTEXT01.pdf.
47. M. K. Matej Gjurkovic, Iva Vukojevic, Mihaela Bosˇnjak, Jan Sˇnajder, "PANDORA Talks: Personality and Demographics on Reddit," 2021, doi: 10.48550/arxiv.2004.04460.
48. Scikit-Learn, "scikit-learn Machine Learning in Python.," N.D. [Online]. Available: https://scikit-learn.org/stable/index.html.
49. Tensorflow, "Tensorflow," N.D. [Online]. https://www.tensorflow.org/.
50. Gensim. "Gensim." https://radimrehurek.com/gensim/ (accessed.
51. H. Face, "Transformers," N.D. [Online]. Available: https://huggingface.co/docs/transformers/index.
52. Pandas, "Pandas," N.D. [Online]. Available: https://pandas.pydata.org/.