VECTORAL PUBLISHING

Research Article: Journal of Computational Social Dynamics Volume: 08

Application of Pre-trained Models (PTMs) in sentiment analysis, news classification, anti-spam detection, and information extraction

Norliza Binti Ahmad

Universiti Teknologi MARA, Kampus Jasin, 77000 Jasin, Melaka, Malaysia

Abstract

The This research aims to investigate the application of Pre-trained Models (PTMs) in Natural Language Processing (NLP), focusing on four key tasks: sentiment analysis, news classification, anti-spam detection, and information extraction. Leveraging PTMs such as BERT, GPT, RoBERTa, and T5, we explore various methodologies tailored for each task. For sentiment analysis, we consider fine-tuning using the IMDb dataset, zero-shot or few-shot learning, and embedding-based approaches that utilize classical classifiers like SVM or Random Forest. In news classification, the study employs fine-tuning on labeled news articles, hierarchical attention to manage longer texts, and transfer learning to adapt models to smaller datasets. For anti-spam detection, the research investigates fine-tuning on spam-specific datasets, anomaly detection techniques, and active learning methods to adapt to the evolving nature of spam. In the domain of information extraction, we engage in Named Entity Recognition (NER), relation extraction, coreference resolution, and template filling to derive structured information from unstructured texts. The advantages of using PTMs include data efficiency, allowing for strong performance with less labeled data; generalization capabilities across different tasks and domains due to their extensive training; and speed, as transfer learning and fine-tuning are usually quicker than building models from the ground up. However, there are challenges to consider: PTMs require significant computational resources, may overfit when applied to small datasets without proper regularization, and offer limited interpretability due to their complex architectures.

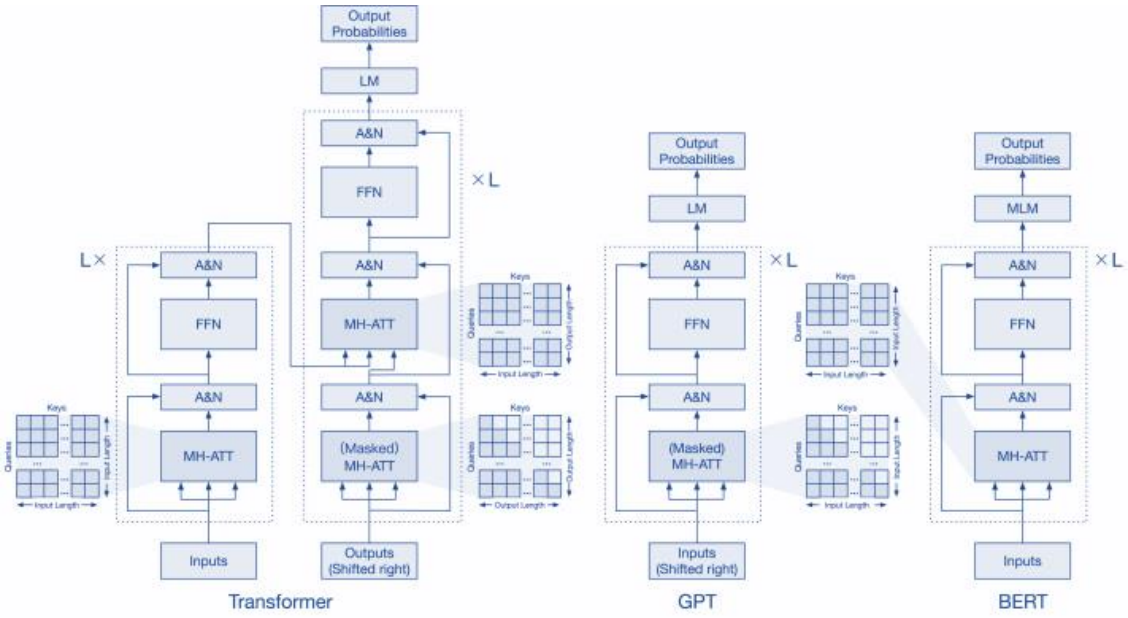
Keywords: *Anti-spam Detection, Fine-tuning, Information Extraction, Pre-trained Models, Sentiment Analysis, Transfer Learning.*

Introduction

Pre-trained Models (PTMs) have brought about significant changes in the field of natural language processing (NLP), reshaping the way researchers and practitioners approach various tasks [1], [2]. In particular, transformer-based architectures like BERT (Bidirectional Encoder Representations from Transformers), GPT (Generative Pre-trained Transformer), RoBERTa (Robustly Optimized BERT Pretraining), and T5 (Text-To-Text Transfer Transformer) stand out for their exceptional performance and versatility. The primary advantage of these architectures

lies in their ability to encode complex language structures through layers of self-attention mechanisms, making it possible to understand context and semantics in a way that was challenging for previous models.

Figure 1. Pre-trained Models (PTMs) ransformer, GPT, and BERT



The inception of transformer architectures can be traced back to the landmark paper "Attention Is All You Need," published by Vaswani et al. in 2017. This paper laid the groundwork for attention-based mechanisms that eliminated the need for recurrent or convolutional layers, traditionally used in sequence modeling tasks. The model introduced a novel way to process sequences in parallel rather than sequentially, thereby dramatically reducing training time while maintaining or even increasing performance on various tasks [3]. The success of this architecture led to the development of a plethora of models that leveraged the transformer's core principles, each fine-tuned for specific NLP challenges.

BERT, introduced by Google in 2018, was a game-changer. Unlike earlier models that processed text from left to right or right to left, BERT used a bidirectional approach. This allowed the model to consider the context from both directions, thereby producing embeddings that are richer and more informative. BERT's training process involves masked language modeling, where certain words in a sentence are masked, and the model is trained to predict them. The ability of BERT to understand context made it especially useful for tasks like question-answering, named entity recognition, and sentiment analysis [4].

Following BERT, OpenAI introduced GPT, which employed a transformer architecture but differed in its training and application. GPT is trained using a left-to-right approach, making it more suitable for generative tasks. The primary focus of GPT is to predict the next word in a sequence, which is in contrast to BERT’s masked language model. Despite this difference in focus, GPT has shown remarkable success in various NLP tasks, from translation to text summarization, and has been widely adopted for chatbot development and other conversational agents [5].

RoBERTa came as an optimized version of BERT, developed by Facebook's AI research team. It differed from BERT mainly in training strategy and data preprocessing. RoBERTa uses dynamic masking rather than static masking, and it's trained on a much larger dataset. These optimizations led to improved performance on several benchmarks, making RoBERTa another influential model in the NLP arena.

T5 is a unique variation in the line of transformer-based models, designed with the philosophy that most NLP tasks can be reframed as text-to-text tasks. Whether it is translation, summarization, or question-answering, T5 approaches them all as a process of converting input text into target text. This unified framework simplifies the process of applying the model to a wide range of tasks without the need for task-specific architectures.

In terms of application, these PTMs have found utility in a broad range of industries and fields. In healthcare, they are used for automatic diagnosis and medical literature mining. In the legal sector, they assist in document review and legal research. In customer service, chatbots powered by these architectures handle queries with increasing accuracy, thereby reducing human workload. In academia and research, these models facilitate advanced text analysis, aiding in everything from data extraction to the development of new NLP techniques.

The generalizability of these models is another crucial feature. Originally trained on massive datasets, these architectures capture the underlying structures and nuances of language, enabling them to be fine-tuned for specific tasks with a relatively small amount of additional data. This saves both time and computational resources, making it easier for small organizations and individual researchers to adopt these models.

The computational cost of training these models is high, requiring specialized hardware and significant energy consumption. There's also an ongoing debate about the interpretability of these models. While they perform exceptionally well, understanding why they make a particular prediction remains difficult, which is a concern in critical applications like healthcare and judiciary. They've excelled in a wide variety of tasks by leveraging their ability to understand complex language structures, making them indispensable tools in today's data-driven world. While challenges remain, especially around computational costs and interpretability, there is no doubt that these models have set a new standard in NLP, expanding the possibilities of what can be achieved with machine understanding of human language [6].

1. Sentiment Analysis:

Fine-tuning pre-trained models for specific tasks such as sentiment analysis is a common practice that leverages the broad knowledge base of a pre-trained model to specialize it for a particular application. Considering BERT (Bidirectional Encoder Representations from Transformers) as an example, and imagine fine-tuning it using the IMDb reviews dataset for sentiment classification. In such a case, a pre-trained BERT model, which has already learned useful language representations from a massive corpus of text, is exposed to the movie reviews in the IMDb dataset [7]. During this fine-tuning phase, the model learns to apply its generalized language understanding abilities to the specific task of categorizing the sentiment of the review as either positive or negative. Essentially, the fine-tuning phase adapts the deep, layered contextual understandings that BERT has developed into focused insights that are relevant to the task at hand. Fine-tuning provides a way to achieve high performance with relatively less data and computational resources compared to training a model from scratch.

Another exciting avenue in the application of Pre-trained Models (PTMs) is zero-shot or few-shot learning. Models like GPT-3 and GPT-4 are so sophisticated in their general language understanding that they can often perform specialized tasks without any fine-tuning. You could ask GPT-3 a straightforward natural language question like, "Is the sentiment of this text positive or negative?" and expect a reasonably accurate response based purely on the model's pre-training. This is particularly advantageous in situations where gathering a large, specialized dataset for training is impractical. It's also a time-saver, as it eliminates the need to fine-tune the model for the specific task. However, it should be noted that while zero-shot and few-shot learning are promising, they may not always yield results as accurate as a fine-tuned model for certain specialized tasks.

Embedding-based approaches offer another method for applying PTMs to tasks like sentiment analysis. In this approach, instead of fine-tuning the model on a specific task, we extract embeddings (vector representations) of text inputs from a pre-trained model like BERT or GPT. These embeddings capture semantic features of the text. Once these embeddings are generated, they can be fed into classical machine learning classifiers like Support Vector Machines (SVM) or Random Forest to perform sentiment classification [8]. The advantage of this approach is that it allows one to leverage both the advanced language understanding capabilities of modern PTMs and the strengths of traditional machine learning algorithms. Embedding-based approaches can be particularly useful when you want a quick, somewhat interpretable method for classifying text without the computational overhead of fine-tuning a large-scale neural network [9].

When comparing these three approaches, each has its own set of advantages and drawbacks. Fine-tuning is computationally expensive but tends to yield highly accurate and task-specific results. Zero-shot or few-shot learning is convenient and quick but may lack the precision of a fine-tuned model [10], [11]. Embedding-based approaches offer a middle ground, combining some of the best features of neural networks and classical machine learning algorithms but potentially lacking the depth of understanding that fine-tuned models can achieve [12]. The choice between these approaches often depends on various factors such as the availability of labeled data, computational resources [13], [14], and the level of accuracy required for the task [15].

In practice, these methods are not mutually exclusive and can often be used in conjunction to achieve optimal results. For example, one could start with zero-shot learning to quickly gauge the sentiment in a set of text data, then move on to fine-tuning a model like BERT for more accurate, nuanced sentiment analysis. Similarly, embedding-based approaches could be used as a preliminary step before deciding whether the computational investment in fine-tuning is justified for a specific task [16].

Despite their capabilities, these methods are not without challenges. For fine-tuning, the risk of overfitting on a small dataset is a concern. Proper regularization techniques and model evaluation are crucial. In the case of zero-shot or few-shot learning, the model's predictions may sometimes be off-mark due to its lack of exposure to task-specific data. Interpretability is a concern for both methods. Embedding-based approaches, on the other hand, might offer slightly more transparency but can suffer from the loss of contextual richness when the embeddings are removed from their original neural network structure and placed into a classical machine learning model.

The rise of PTMs in NLP has broadened the range of techniques available for tasks like sentiment analysis. Whether through fine-tuning, zero-shot learning, or embedding-based approaches, these models offer robust, flexible methods for converting text into actionable insights. As ongoing research continues to explore the capabilities and limitations of these various approaches, it is likely that we will see even more advanced techniques and hybrid models that combine the strengths of each method to achieve increasingly accurate and nuanced language understanding.

2. News Classification:

Fine-tuning pre-trained models on a dataset of news articles for classification into various categories is a compelling use-case that demonstrates the versatility of PTMs like BERT, GPT, and others [17], [18]. News articles are a complex type of text, often filled with nuanced language, embedded clauses, and domain-specific vocabulary. A pre-trained model like BERT, which has been initially trained on a large corpus of text from diverse sources, can be adapted to understand the subtleties of news articles by continuing its training on a labeled dataset. The categories can range from "Politics" and "Business" to "Entertainment" and "Technology" [19]. By fine-tuning the pre-trained model on this specialized dataset, its internal parameters are adjusted to minimize the classification error, thereby equipping the model with the ability to categorize news articles with high accuracy. Fine-tuning is generally effective because it allows the model to capitalize on its general understanding of language while tailoring its abilities to meet the specific needs of news categorization [20].

For handling longer news articles, hierarchical attention mechanisms offer an intriguing approach. In a standard flat architecture, the model considers each word in the article equally to produce a final representation for classification. However, this may not be the most effective strategy when articles are long and contain multiple sentences or paragraphs, each with varying degrees of relevance to the topic. Hierarchical models work in a two-step fashion: first, sentence representations are created by encoding each sentence in the article; then, these sentence representations are themselves encoded to produce an article-level representation. Attention mechanisms can be applied at both levels, allowing the model to focus on the most relevant sentences when classifying the article. This hierarchical structure mimics the way humans often read long articles, skimming through to focus more on the sentences or sections that appear most relevant [21].

Transfer learning represents another practical approach to news article classification, particularly beneficial when the available labeled dataset for news is small. Here, a model is initially trained on a large dataset from a different domain that has some relevance to news articles. The model learns to extract useful features and understand complex language structures during this phase. Once this is accomplished, the model is then fine-tuned on the smaller dataset of news articles. The idea is that the features learned during the initial training phase can be "transferred" to make the fine-tuning process more effective. Transfer learning is a strategy that helps alleviate the limitations of having a small dataset for the task at hand, as it allows the model to generalize well to new, unseen articles in the same category.

Choosing between fine-tuning, hierarchical attention, and transfer learning often depends on the specific requirements of the task, the nature of the dataset, and the computational resources available. Fine-tuning is generally a strong choice when a sufficiently large labeled

dataset is available and when the categories in question are numerous and nuanced. Hierarchical attention mechanisms shine when dealing with long, intricate articles where understanding the context at multiple levels is crucial for accurate classification. Transfer learning is particularly useful when the dataset for the specific task is limited, but a larger, related dataset is available for initial training [22].

Each of these methods also has its own set of challenges. Fine-tuning may result in overfitting if the news dataset is small and not sufficiently diverse. Hierarchical models can be computationally intensive and may require more sophisticated hardware. Transfer learning risks the possibility that the knowledge transferred from the initial domain may not be perfectly aligned with the specific requirements of news classification, necessitating additional steps to ensure relevance [23].

In practical applications, it's not uncommon to see combinations of these techniques for optimal performance. For instance, one could use transfer learning to pre-train a model on a large corpus and then fine-tune it on a specific news dataset [24], [25]. Furthermore, hierarchical attention could be incorporated into this model to better handle long articles. Through continual research and development, these methodologies are being refined and adapted, offering increasingly sophisticated tools for the complex task of news article classification [26].

3. Anti-spam Detection:

Fine-tuning pre-trained models on a dataset comprising spam and non-spam messages has proven effective for spam detection tasks. The essence of spam often lies in its subtle differences from legitimate messages, and these nuances require a model to have a deep understanding of language to effectively distinguish between the two. For example, a pre-trained model like BERT, which has a strong understanding of language semantics and structure, can be fine-tuned on a labeled dataset where spam and non-spam messages are explicitly tagged. This process adjusts the model's internal parameters so that it can better capture the particular features, patterns, and structures commonly found in spam messages. By doing so, the model becomes adept at identifying spam content, thereby enhancing the efficacy of spam filters.

Anomaly detection is another technique that lends itself well to spam detection. In this approach, pre-trained embeddings from models like BERT or GPT are used to generate vector representations of messages [27], [28]. These vectors capture the semantic essence of each message and allow the system to understand the "usual" structure or pattern of legitimate, non-spam messages. By analyzing these embeddings, the model can identify deviations from the norm. Messages that deviate substantially from this pattern can be flagged as potential spam. This is especially useful for capturing new types of spam that may not yet have been labeled in existing datasets, thereby offering a level of adaptability and responsiveness [29].

Active learning is a technique that addresses the evolving nature of spam. Spam strategies frequently change as spammers try to outwit existing filters. To keep pace, an active learning approach can be employed. In this setup, the model initially makes predictions based on its current understanding. Messages that the model is least confident about are flagged for review. Human experts can then label these instances as either spam or non-spam. The model is retrained on this enriched dataset, incorporating these new instances to improve its

performance. The cycle is repeated iteratively, allowing the model to adapt to new forms of spam as they emerge.

Selecting between fine-tuning, anomaly detection, and active learning is often dependent on the specific needs of the system and the characteristics of the spam being targeted. Fine-tuning is a powerful approach when a large, labeled dataset is available and when the features of spam are well-understood. Anomaly detection is more suited for situations where the nature of spam is evolving quickly and the system needs to be able to adapt without waiting for new labels. Active learning is particularly beneficial when human expertise is available for iterative labeling, and when the cost of false positives and false negatives is high, necessitating continuous model improvement [30]. Each approach has its own challenges and considerations. Fine-tuning can be data-intensive and may require periodic retraining as the nature of spam evolves. Anomaly detection can sometimes produce false positives, flagging legitimate messages as spam if they deviate from the 'usual' pattern. Active learning is labor-intensive and requires a commitment to ongoing human involvement for labeling.

In many practical implementations, a combination of these methods is often employed to maximize performance. For example, a system may use fine-tuning to establish a strong baseline model, employ anomaly detection to catch new types of spam, and then use active learning to continuously refine the model over time. As spam detection is a dynamic problem, the multiplicity of these approaches offers a robust, adaptable, and effective way to tackle it.

4. Information Extraction:

Named Entity Recognition (NER) is a critical task in natural language processing that involves identifying entities like persons, organizations, and locations within a given text. Pre-trained models like BERT or RoBERTa can be fine-tuned specifically on NER datasets to improve their performance in this task. Fine-tuning helps the model to recognize the unique lexical and syntactic cues associated with different types of entities. For example, the word "Street" following a proper noun might indicate a location, or the use of "Inc." or "Corp." may denote an organization. By adjusting the model parameters based on a labeled NER dataset, the model gains a specialized ability to identify such entities in a wide range of texts [31].

Relation extraction goes hand-in-hand with NER and focuses on identifying the relationships between the recognized entities. Once entities are identified, models can be trained or fine-tuned to understand the type of relationship between them. For instance, a sentence like "Barack Obama was born in Hawaii" contains the entities "Barack Obama" and "Hawaii," and the relationship "was born in" between them. Various machine learning models or even rule-based systems can operate on top of the entity recognition layer to extract these relationships, offering a more nuanced understanding of the text [32].

Coreference resolution is another task that contributes to the understanding of text by determining which words or phrases refer to the same entity across sentences or within the same sentence. For instance, in the sentence "Jane said she would come," the words "Jane" and "she" refer to the same entity. Models specialized in coreference resolution can make these connections, which is essential for tasks like document summarization, question answering, and many others. Some pre-trained models can be adapted for coreference resolution, although this task often requires specialized architectures and training regimes due to its complexity.

Template filling serves the purpose of extracting structured information from unstructured text. For example, one could create a template for "CEO-Company" pairs and train a model to fill in this template by reading through news articles or press releases. This is particularly useful for information retrieval systems or databases where structured information is desired. Pre-trained models fine-tuned on a dataset of labeled examples can perform this task quite effectively. They can identify the entities that fit into the template and extract them, converting a piece of unstructured text into a structured data entry [33].

The choice between these various tasks and techniques—NER, relation extraction, coreference resolution, and template filling—depends on the specific objectives and the nature of the data you're working with. NER is often the foundational step for many text analytics processes, while relation extraction can add layers of understanding that are critical for more sophisticated analyses. Coreference resolution is necessary for tasks that require a deep understanding of the text's context and narrative, and template filling is crucial when the goal is to convert textual information into a structured format [34].

Each task comes with its own challenges. Fine-tuning for NER can be resource-intensive, and the model may need periodic updates to adapt to new types of entities or naming conventions. Relation extraction faces the challenge of handling ambiguous relationships, and it often requires a labeled dataset that marks not just entities but also the relations between them. Coreference resolution is complicated by linguistic phenomena like anaphora and cataphora, where the reference may appear before or after the entity it refers to. Template filling, on the other hand, must deal with the variability in how the same information is expressed in different texts [35], [36].

Real-world applications often combine these techniques for maximum effectiveness. For example, a business intelligence application may first employ NER to identify companies and key personnel, then use relation extraction to determine the relationships between them, followed by template filling to populate a database with these structured relationships [37]. As NLP technologies continue to advance, these techniques are expected to become even more effective and integrated, enabling richer and more accurate understanding of text.

Advantages of Using PTMs:

Data efficiency is a notable advantage of using pre-trained models (PTMs) like BERT, GPT, or RoBERTa in natural language processing tasks. These models come pre-trained on massive datasets, capturing a wide array of language structures, patterns, and semantics. When fine-tuned on a specific task, they often require less labeled data to achieve strong performance compared to models trained from scratch. This is because they already possess a foundational understanding of language and merely need to adapt to the nuances of the specific task at hand. For example, if you were to use a PTM for a sentiment analysis task, the model could effectively understand the sentiment of text even with a relatively small dataset for fine-tuning. This data efficiency is particularly beneficial in scenarios where obtaining large quantities of labeled data is challenging or expensive [38].

The generalization ability of PTMs is another compelling feature. Because these models are trained on diverse and extensive datasets, they have a broad understanding of language. This extensive pre-training enables them to generalize well across various tasks and domains. For instance, a model pre-trained on a corpus that includes literature, websites, and scientific

articles would be versatile enough to be fine-tuned for tasks ranging from text summarization to medical diagnosis interpretation. This makes PTMs highly flexible tools that can be adapted for a wide range of applications, thereby maximizing the return on the computational and financial investment required for their initial training [39]

Speed is yet another benefit that comes with using PTMs, particularly when leveraging transfer learning and fine-tuning methods. Training a deep learning model from scratch demands significant computational resources and time, especially if the model architecture is complex and the dataset is large. In contrast, fine-tuning a pre-trained model for a specific task can be considerably faster. This is because the pre-trained model has already learned many of the foundational elements of language and only needs to adjust its existing knowledge to the specific task [40]. The fine-tuning process thus involves fewer iterations and can converge to a good solution more quickly.

Choosing between data efficiency, generalization, and speed is often a matter of identifying the most pressing needs of a specific project. For tasks where labeled data is scarce, the data efficiency of PTMs may be the most appealing aspect. In cases where a model needs to perform well on a range of tasks, the generalization capabilities of PTMs could be the primary focus. And in scenarios where time is of the essence, the speed advantages of using pre-trained models could take precedence.

Data efficiency, while advantageous, can sometimes lead to overfitting if the fine-tuning data is too sparse or not representative. Generalization, although generally a strength of PTMs, can falter if the model encounters data or tasks that are too far removed from its original training set. And while speed is a benefit, fine-tuning still requires computational resources, particularly for large models. These challenges often necessitate a well-planned approach that balances the advantages and limitations of using PTMs for specific applications [41].

Conclusion:

Computational requirements are a significant consideration when working with pre-trained models (PTMs) like BERT, GPT, or RoBERTa. These models, especially the larger versions, demand substantial computational resources for both training and inference. The training phase requires powerful hardware, typically multiple GPUs or TPUs, along with considerable storage space to house the massive datasets and model parameters. The inference phase, although generally less resource-intensive than training, still requires a capable setup to ensure timely responses. These computational requirements often mean increased costs and can limit the accessibility of these models for smaller organizations or individual researchers who may not have the necessary resources.

Overfitting on small datasets is another challenge that arises with PTMs. These models are complex, with millions or even billions of parameters, and this complexity can lead to overfitting when the models are fine-tuned on small, task-specific datasets. Overfitting occurs when the model learns the training data too well, capturing its noise rather than its underlying pattern. As a result, it performs poorly on new, unseen data. To mitigate this, careful regularization techniques must be applied during the fine-tuning process. Techniques such as dropout, weight decay, or reducing the model size can help prevent overfitting, but they add another layer of complexity to the model training and selection process.

Interpretability remains a major challenge with PTMs. While these models are exceptionally good at a wide range of tasks, understanding why they make a particular decision is difficult due to their complex architectures. This is a significant issue in applications where understanding the reasoning behind decisions is crucial, such as healthcare, finance, or legal settings. Various techniques like attention heatmaps, LIME (Local Interpretable Model-agnostic Explanations), and SHAP (SHapley Additive exPlanations) have been proposed to shed light on the decision-making processes of these models, but interpretability is still an active area of research [42] [43].

Choosing between computational efficiency, risk of overfitting, and interpretability involves a careful balancing act depending on the specific requirements of a project. If computational resources are a constraint, one may opt for smaller versions of PTMs, sacrificing some performance for the sake of feasibility. When working with small datasets, special attention must be given to regularization techniques to prevent overfitting. And if interpretability is a concern, one might explore hybrid models or specialized interpretability techniques, even if they add to the complexity or slightly reduce the performance.

However, these challenges should be viewed in the context of the immense benefits that PTMs offer [44], [45]. Despite the computational costs, their data efficiency and ability to generalize across tasks often make them a cost-effective choice in the long run. Even with the risk of overfitting, their performance on small datasets often surpasses that of models trained from scratch. And while interpretability remains an issue, the sheer effectiveness of PTMs on complex tasks can make them indispensable, even as efforts to improve their transparency continue. Therefore, while it's essential to be mindful of these challenges, they are often outweighed by the considerable advantages PTMs bring to the table in various NLP applications.

References

[1] J. He, J. Zhai, T. Antunes, H. Wang, F. Luo, and S. Shi, “FasterMoE: modeling and optimizing training of large-scale dynamic pre-trained models,” *Proceedings of the 27th*, 2022.

[2] K. Kurita, P. Michel, and G. Neubig, “Weight Poisoning Attacks on Pre-trained Models,” *arXiv [cs.LG]*, 14-Apr-2020.

[3] Y. Huang *et al.*, “Behavior-driven query similarity prediction based on pre-trained language models for e-commerce search,” 2023.

[4] R. S. S. Dittakavi, “Evaluating the Efficiency and Limitations of Configuration Strategies in Hybrid Cloud Environments,” *International Journal of Intelligent Automation and Computing*, vol. 5, no. 2, pp. 29–45, 2022.

[5] X. Jiang, Z. Zheng, C. Lyu, L. Li, and L. Lyu, “TreeBERT: A tree-based pre-trained model for programming language,” in *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, 27--30 Jul 2021, vol. 161, pp. 54–63.

[6] J. Gesi, H. Wang, B. Wang, A. Truelove, J. Park, and I. Ahmed, “Out of Time: A Case Study of Using Team and Modification Representation Learning for Improving Bug Report Resolution Time Prediction in Ebay,” *Available at SSRN 4571372*.

[7] J. Yang, G. Xiao, Y. Shen, W. Jiang, and X. Hu, “A survey of knowledge enhanced pre-trained models,” *arXiv preprint arXiv*, 2021.

[8] S. Khanna, “Brain Tumor Segmentation Using Deep Transfer Learning Models on The Cancer Genome Atlas (TCGA) Dataset,” *Sage Science Review of Applied Machine Learning*, vol. 2, no. 2, pp. 48–56, 2019.

[9] H. Vijayakumar, “Business Value Impact of AI-Powered Service Operations (AIServiceOps),” *Available at SSRN 4396170*, 2023.
[10] Z. Yang, J. Shi, J. He, and D. Lo, “Natural attack for pre-trained models of code,” *of the 44th International Conference on …*, 2022.
[11] V. Kumar, A. Choudhary, and E. Cho, “Data Augmentation using Pre-trained Transformer Models,” *arXiv [cs.CL]*, 04-Mar-2020.
[12] S. Khanna, “COMPUTERIZED REASONING AND ITS APPLICATION IN DIFFERENT AREAS,” *NATIONAL JOURNAL OF ARTS, COMMERCE & SCIENTIFIC RESEARCH REVIEW*, vol. 4, no. 1, pp. 6–21, 2017.
[13] A. Karmakar and R. Robbes, “What do pre-trained code models know about code?,” in *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2021, pp. 1332–1336.
[14] R. Hennequin, A. Khlif, and F. Voituret, “Spleeter: a fast and efficient music source separation tool with pre-trained models,” *Journal of Open Source*, 2020.
[15] S. Khanna and S. Srivastava, “Patient-Centric Ethical Frameworks for Privacy, Transparency, and Bias Awareness in Deep Learning-Based Medical Systems,” *Applied Research in Artificial Intelligence and Cloud Computing*, vol. 3, no. 1, pp. 16–35, 2020.
[16] J. Gesi, X. Shen, Y. Geng, Q. Chen, and I. Ahmed, “Leveraging Feature Bias for Scalable Misprediction Explanation of Machine Learning Models,” in *Proceedings of the 45th International Conference on Software Engineering (ICSE)*, 2023.
[17] X. Wang *et al.*, “Large-scale Multi-modal Pre-trained Models: A Comprehensive Survey,” *Machine Intelligence Research*, vol. 20, no. 4, pp. 447–482, Aug. 2023.
[18] Z. Zeng, H. Tan, H. Zhang, J. Li, Y. Zhang, and L. Zhang, “An extensive study on pre-trained models for program understanding and generation,” in *Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis*, Virtual, South Korea, 2022, pp. 39–51.
[19] H. Vijayakumar, “The Impact of AI-Innovations and Private AI-Investment on U.S. Economic Growth: An Empirical Analysis,” *Reviews of Contemporary Business Analytics*, vol. 4, no. 1, pp. 14–32, 2021.
[20] R. S. S. Dittakavi, “Dimensionality Reduction Based Intrusion Detection System in Cloud Computing Environment Using Machine Learning,” *International Journal of Information and Cybersecurity*, vol. 6, no. 1, pp. 62–81.
[21] J. Gesi *et al.*, “Code smells in machine learning systems,” *arXiv preprint arXiv:2203.00803*, 2022.
[22] P. Marcelino, “Transfer learning from pre-trained models,” *Towards data science*, 2018.
[23] H. Vijayakumar, A. Seetharaman, and K. Maddulety, “Impact of AIServiceOps on Organizational Resilience,” 2023, pp. 314–319.
[24] K. You, Y. Liu, J. Wang, and M. Long, “Logme: Practical assessment of pre-trained models for transfer learning,” *International Conference on*, 2021.
[25] Y. Du, Z. Liu, J. Li, and W. X. Zhao, “A survey of vision-language pre-trained models,” *arXiv preprint arXiv:2202.10936*, 2022.
[26] S. Khanna and S. Srivastava, “AI Governance in Healthcare: Explainability Standards, Safety Protocols, and Human-AI Interactions Dynamics in Contemporary Medical AI Systems,” *Empirical Quests for Management Essences*, vol. 1, no. 1, pp. 130–143, 2021.
[27] R. Wang *et al.*, “K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters,” *arXiv [cs.CL]*, 05-Feb-2020.
[28] Z. Feng, D. Guo, D. Tang, N. Duan, and X. Feng, “Codebert: A pre-trained model for programming and natural languages,” *arXiv preprint arXiv*, 2020.

[29] S. Khanna, “A Review of AI Devices in Cancer Radiology for Breast and Lung Imaging and Diagnosis,” *International Journal of Applied Health Care Analytics*, vol. 5, no. 12, pp. 1–15, 2020.
[30] A. Groce *et al.*, “Evaluating and improving static analysis tools via differential mutation analysis,” in *2021 IEEE 21st International Conference on Software Quality, Reliability and Security (QRS)*, 2021, pp. 207–218.
[31] H. Chen *et al.*, “Pre-Trained Image Processing Transformer,” *arXiv [cs.CV]*, pp. 12299–12310, 01-Dec-2020.
[32] H. Vijayakumar, “Revolutionizing Customer Experience with AI: A Path to Increase Revenue Growth Rate,” 2023, pp. 1–6.
[33] H. Vijayakumar, “Unlocking Business Value with AI-Driven End User Experience Management (EUEM),” in *2023 5th International Conference on Management Science and Industrial Engineering*, 2023, pp. 129–135.
[34] S. Khanna, “Identifying Privacy Vulnerabilities in Key Stages of Computer Vision, Natural Language Processing, and Voice Processing Systems,” *International Journal of Business Intelligence and Big Data Analytics*, vol. 4, no. 1, pp. 1–11, 2021.
[35] Y. Cui, W. Che, T. Liu, B. Qin, S. Wang, and G. Hu, “Revisiting Pre-Trained Models for Chinese Natural Language Processing,” *arXiv [cs.CL]*, 29-Apr-2020.
[36] Q. Zhu and J. Luo, “Generative Pre-Trained Transformer for Design Concept Generation: An Exploration,” *Proceedings of the Design Society*, vol. 2, pp. 1825–1834, May 2022.
[37] H. Vijayakumar, “Impact of AI-Blockchain Adoption on Annual Revenue Growth: An Empirical Analysis of Small and Medium-sized Enterprises in the United States,” *International Journal of Business Intelligence and Big Data Analytics*, vol. 4, no. 1, pp. 12–21, 2021.
[38] S. Khanna, “EXAMINATION AND PERFORMANCE EVALUATION OF WIRELESS SENSOR NETWORK WITH VARIOUS ROUTING PROTOCOLS,” *International Journal of Engineering & Science Research*, vol. 6, no. 12, pp. 285–291, 2016.
[39] R. S. S. Dittakavi, “An Extensive Exploration of Techniques for Resource and Cost Management in Contemporary Cloud Computing Environments,” *Applied Research in Artificial Intelligence and Cloud Computing*, vol. 4, no. 1, pp. 45–61, Feb. 2021.
[40] J. Gesi, J. Li, and I. Ahmed, “An empirical examination of the impact of bias on just-in-time defect prediction,” in *Proceedings of the 15th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 2021, pp. 1–12.
[41] R. S. S. Dittakavi, “Deep Learning-Based Prediction of CPU and Memory Consumption for Cost-Efficient Cloud Resource Allocation,” *Sage Science Review of Applied Machine Learning*, vol. 4, no. 1, pp. 45–58, 2021.
[42] F. Jirigesi, A. Truelove, and F. Yazdani, “Code Clone Detection Using Representation Learning.”
[43] F. N. U. Jirigesi, “Personalized Web Services Interface Design Using Interactive Computational Search.” 2017.
[44] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, “Pre-trained models for natural language processing: A survey,” *Science China*, 2020.
[45] X. Han *et al.*, “Pre-trained models: Past, present and future,” *AI Open*, 2021.