Characterizing Text Datasets with Psycholinguistic Features

Marcio Monteiro, Charu James, Marius Kloft, Sophie Fellenz
RPTU Kaiserslautern-Landau, Germany
marcio.monteiro@cs.rptu.de, {charu, kloft, fellenz}@cs.uni-kl.de

Abstract

Fine-tuning pretrained language models on task-specific data is a common practice in Natural Language Processing (NLP) applications. However, the number of pretrained models available to choose from can be very large, and it remains unclear how to select the optimal model without spending considerable amounts of computational resources, especially for the text domain. To address this problem, we introduce PsyMatrix, a novel framework designed to efficiently characterize text datasets. PsyMatrix evaluates multiple dimensions of text and discourse, producing interpretable, low-dimensional embeddings. Our framework has been tested using a meta-dataset repository that includes the performance of 24 pretrained large language models fine-tuned across 146 classification datasets. Using the proposed embeddings, we successfully developed a meta-learning system capable of recommending the most effective pretrained models (optimal and near-optimal) for fine-tuning on new datasets.

1 Introduction

Pre-training a language model on a large, diverse, and unlabeled corpus, then fine-tuning it with task-specific data has proven to be highly effective for enhancing performance of natural language processing (NLP) applications (Radford and Narasimhan, 2018). Since then, numerous pretrained large language models (LLM) have been released to the public, each pretrained on different corpora with varying sizes, and using different architectures. Although the prospect of a universal pretrained model (PTM) that excels across all NLP tasks is attractive, evidence (still) indicates that no single model performs optimally in every scenario (Wolpert, 1996; Lorena et al., 2019; Arango et al., 2024). This poses a challenge for machine learning practitioners, who must select a PTM to fine-tune for a task-specific dataset. Performing an exhaustive search over all possible candidates can be very time and resource consuming. In practice, this is not realistic.

The performance of PTMs often fluctuates depending on some characteristics of the target dataset (Schaffer, 1994), usually referred to as meta-features (Rivolli et al., 2022). For example, a model that performs exceptionally well on carefully written news articles may encounter difficulties with the brevity and slang commonly found in social media posts (Zheng and Yang, 2019; Shushkevich et al., 2022; Roussinov and Sharoff, 2023). Therefore, the challenge is to decide, for a particular dataset, which PTM is expected to perform the best after fine-tuning.

One low-cost approach could be to search for public benchmarks and check which PTM performed the best for similar datasets. However, defining such dataset similarity in an objective manner, specially for text datasets, is very challenging.

To address this, we propose a novel framework named PsyMatrix, which characterizes text datasets by analyzing multiple aspects of text and discourse. These range from simple part-of-speech (POS) statistics to more complex and deep psycholinguistic features (Barnwal and Tiwary, 2017). Our framework generate dataset embeddings that are both interpretable and low-dimensional, providing a deeper understanding of the intricacies inherent in different datasets. Figure 1 illustrates the core components of the framework. The code of the framework is open-source and has been made publicly available[1].

To sum up, our contributions are as follows:

- Introduction of PsyMatrix: a framework to characterize text datasets by generating interpretable and low-dimensional embeddings that capture several levels of language and discourse.

[1] https://github.com/contemmcm/psymatrix

Findings of the Association for Computational Linguistics: EACL 2024, pages 14977–14990

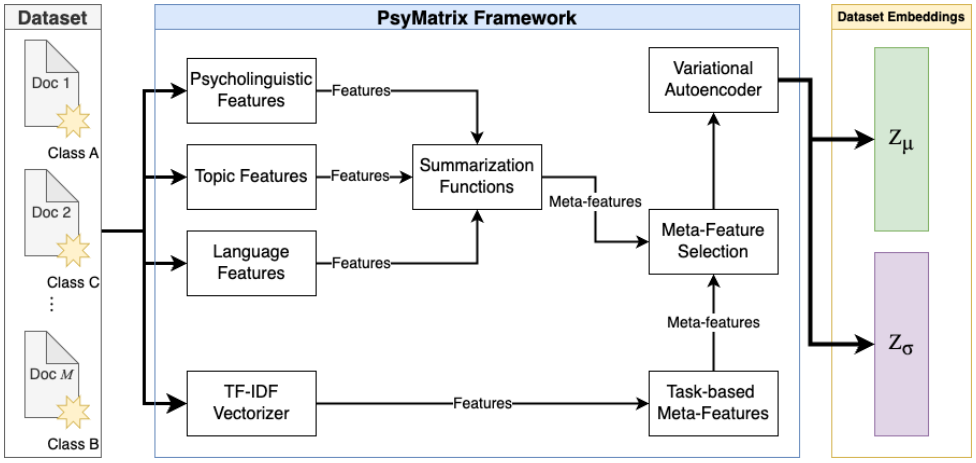


Figure 1: PsyMatrix: Text classification datasets (left) are transformed via different feature extractors (middle left). The features are summarized across documents (middle) and a subset of meta-features is selected (middle right). A variational autoencoder compresses the meta-features to result in the final PsyMatrix dataset embeddings (right).

- Recommendation System Potential: the framework's applicability in creating a recommendation system for optimal or nea-optimal pretrained model selection for new datasets.

- Validation Across Datasets: Fine-tuning and evaluation of 24 pretrained large language models on 146 public text classification datasets to validate the framework.

The paper is organized as follows: Section 2 reviews existing methods for characterizing datasets in general and for the problem of selecting a pretrained model for fine-tuning on task-specific data. Section 3 provides a detailed description of the framework, including the psycholinguistic dimensions and complexity measures used. Section 4 presents and analyzes the findings from fine-tuning models on 146 datasets and the predictive performance of the provided embeddings, presenting practical applications of PsyMatrix in guiding model selection and its broader implications in machine learning. Finally, Section 5 summarizes the findings, contributions, and potential future work.

2 Related Work

The process of selecting an optimal PTM for a specific dataset typically involves trial and error. Practitioners usually start by choosing a few potential models based on prior knowledge and experience (Alzahrani et al., 2022; Ren et al., 2023; Daban et al., 2023; Malic et al., 2023; Qiu et al., 2022). These models are then fine-tuned on the target dataset and have their performance compared, allowing for an assessment of how well each model adapts to the unique characteristics of the data.

However, the result of this approach can be limited by the assumptions made during initial model pre-selection, which may not always be valid for the dataset in question, which might lead to suboptimal outcomes. This mismatch underscores the need for a refined selection strategy that takes into account the unique features of the dataset, promoting a more effective model selection process.

2.1 Characterization of Datasets

Recognizing specific characteristics or meta-features of the target dataset can considerably refine the selection of potential models. This approach helps eliminate models that historically underperform on similar types of data, emphasizing the importance of accurate meta-feature identification and use. By pinpointing critical meta-features, we can more reliably forecast the performance of machine learning algorithms that are tailored to particular dataset profiles (Brazdil et al., 2022).

Researchers have proposed a wide range of meta-features to characterize datasets in general, in order to aid in the model selection process, spanning from basic statistical attributes like mean and variance, to more advanced measures rooted in complexity and information theory (Lindner and Studer, 1999; Sohn, 1999; Bensusan et al., 2000; Peng et al., 2002; Lorena et al., 2019; Rivolli et al., 2022). Unfortunately, they are all designed for tabular datasets, making them not applicable to text datasets directly. Notably, efforts have been made to adapt meta-feature extraction to image datasets,

demonstrating some success in this area (Edwards and Storkey, 2017; Jomaa et al., 2021).

To our knowledge, the only research addressing specifically text dataset characterization is by Simig et al. (2022), who introduce the Text Characterization Toolkit (TCT). TCT utilizes several existing metrics from the Coh-Metrix toolkit to evaluate the documents in a given dataset, performing various statistical analyses on the extracted features (McNamara et al., 2014). Although the effectiveness of these metrics was not quantified objectively, it contributes to a better understanding of text datasets. Our framework also uses Coh-Metrix as one source of features, but is not limited to it, employing other similar tools as well.

2.2 Pretrained Model Selection

Other approaches have been proposed to address the pretrained model selection problem, using various strategies and architectures (Arango et al., 2024; Li et al., 2023; Bolya et al., 2021; Tang et al., 2024). While these contributions have significantly advanced the field, their focus has predominantly been on image-related tasks (or at least only demonstrated on it). This leaves a gap in methodologies specifically designed for text-based applications. Our research aims to bridge this gap. To the best of our knowledge, this is the first work *specifically* tailored to the text domain.

3 The PsyMatrix Framework

The PsyMatrix framework, as depicted in Figure 1, is designed to synthesize text datasets into a manageable form (i.e., dataset embeddings) using a series of feature extractors and dimensionality reduction techniques. These tools combined, not only capture the essence of the documents, but also facilitate a deeper understanding of underlying patterns and relationships. This way, the PsyMatrix Framework aims to provide dataset embeddings that can be employed as performance predictors for different downstream tasks. This section details the components of the PsyMatrix Framework, illustrating how each module contributes to the overall goal.

3.1 Problem Statement

A dataset $\mathcal{D}$ consists of M documents, each represented by a feature vector of size N, extracted through the function Ψ. This function maps the set of documents to a feature matrix $\mathbf{X}$, where each row corresponds to a document's feature vector. Thus, we express the feature matrix for $\mathcal{D}$ as:

$$\mathbf{X} = \Psi(\mathcal{D}) \in \mathbb{R}^{M \times N} \tag{1}$$

To characterize the entire dataset $\mathcal{D}$, we apply the embedding function Φ, defined in Eq.(2), which transforms $\mathbf{X}$ into a reduced-dimensional representation vector $\mathbf{x}_\phi$ of size K (where $K \ll M \times N$):

$$\mathbf{x}_\phi = \Phi(\mathbf{X}) \in \mathbb{R}^K \tag{2}$$

The objective is to develop an embedding that can effectively serve as a performance predictor for various tasks across different text datasets. To this end, let the matrix Y_{ij} be a meta-dataset containing the actual performances of each model i when fine-tuned on each dataset $\mathcal{D}_j$. Then, let η_i be a surrogate function estimating this performance:

$$\hat{Y}_{ij} = \eta_i(\Phi(\Psi(\mathcal{D}_j)) \tag{3}$$

The problem is thus to find a pipeline $\{\Psi, \Phi, \eta\}$ such that the estimation error across all models and datasets is minimized:

$$\min_{\Psi,\Phi,\eta} \sum_{i=1}^{M}\sum_{j=1}^{D}\left(Y_{ij} - \hat{Y}_{ij}\right)^2 \tag{4}$$

Subsequent sections will discuss empirical strategies to address this optimization challenge.

3.2 Psycholinguistic features

Some texts are easier to read than others. For instance, straightforward sentences with common vocabulary, as those found in children books, are generally simpler for humans to process than complex texts with specialized terminology, like legal contracts or scientific articles. We hypothesize that language models experience similar challenges, which likely affects their performance on various tasks. The key challenge, then, is to find a method for measuring the complexity of text.

Several tools have been proposed to quantify the complexity of text. In this work, we combine features from the following tools as a function ψ_{psy}:

- TextStat[2] (Bansal and Aggarwal, 2024)
- TAACO[3] (Crossley et al., 2016)

[2] https://textstat.org/

[3] https://www.linguisticanalysistools.org/taaco.html

- Coh-Metrix[4] (Graesser et al., 2004, 2011)

Those tools provide sophisticated features of texts, which are usually referred to as psychologists features. Together, they provide more than 300 features. To mention a few:

- Gunning fog index: estimates the number of years in formal education that a person needs to clearly understand the text (Gunning, 1952).
- Rix readability index: it measures readability based on the number of long words in relation to the total number of sentences. A higher score indicates that a text is more complex and potentially harder to understand, while a lower score suggests easier readability (Anderson, 1983).
- Word frequency for content words: measures the frequency of content words (e.g., nouns, verbs, adjectives, adverbs) in the CELEX2 reference corpus (Baayen et al., 1995).
- Hypernyms for verbs nouns and verbs: estimates the specificity of words by the number of its hypernyms in WordNet (Fellbaum, 1998).
- Type-Token Ratio (TTR): quantifies the lexical diversity within a text. It is calculated dividing the number of unique words (types) by the total number of words (tokens). A high TTR indicates a large proportion of unique words within the text. This can make the text more complex and challenging to understand, as it requires the reader to process and integrate a higher number of unique words into the context (Templin, 1957).

For more details on each feature, we encourage the reader to check the official documentation of each tool.

3.3 Topic-based features

Another hypothesis is that language models struggle more with some topics than others, similar to humans. Hence, determining the range of topics covered in a dataset might be beneficial for predicting the performance of a fine-tuned model.

Topic modeling is a branch of unsupervised machine learning that aims to discover the abstract "topics" that occur in a collection of documents. Latent Dirichlet Allocation (LDA) is one of the most popular topic modeling techniques, due to its probabilistic foundations and flexibility (Blei et al., 2003). LDA models each document as a mixture of topics. It outputs a topic distribution vector for each document, where each element of the vector represents the probability of the document to contain a particular topic, hence another source of features for the documents.

However, for LDA to be effective across various datasets, it needs to be trained on a large and diverse set of documents, and the number of topics should be large enough as well. This ensures that the topics that LDA identifies are likely to appear across the majority of documents in any target dataset. These topic features are extracted by ψ_{topic}.

3.4 Language-based features

PsyMatrix also determines the proportion of different languages in a dataset. Although the main purpose of the framework (for now) is to deal with English datasets, we have noticed that, on public datasets (specially those based on user reviews), it is not uncommon to find documents written in different languages. This can also impact the performance of fine-tuned language models that were not pretrained on a multi-language corpus.

Given this, the languages identified within the documents are incorporated as additional features within the framework, one feature per language. This approach helps to enhance model robustness by acknowledging and adapting to the multilingual nature of real-world data. ψ_{language} extracts the language-based features.

In this work, we used a tool called *language-detection*[5] that can identify 55 different languages (hence 55 features) using a naive Bayesian approach and trained on Wikipedia. This tool has a reported accuracy of over 99% for these languages.

3.5 Meta-features summarizer

The feature matrix X is now constructed as $X = \psi_{\text{psy}}(D) + \psi_{\text{topic}}(D) + \psi_{\text{language}}(D)$ where $+$ denotes concatenation. The next step involves converting the extracted document features into dataset meta-features by a function $\phi(X)$. Meta-features are secondary attributes generated from the primary features, designed to capture various statistical aspects of their distribution among documents of a

[4] http://cohmetrix-new.memphis.edu/home

[5] https://pypi.org/project/langdetect/

dataset. To accomplish this, we have used summarizing functions.

Key meta-features include common summarizing functions like the mean, standard deviation, minimum and maximum value, which provide variability and range insights. Other significant meta-features include the mode, skewness, kurtosis, first, second, and third quartile, among others. In total, we have employed 20 summary functions to generate meta-features, which enable a more nuanced understanding of the dataset's underlying patterns and characteristics. The complete list of the summarizing functions can be found in the Appendix.

3.6 Task-Specific Meta-Features (Classification)

For classification tasks, not only is the content of documents important, but also how well these documents match their labels. Take the *20 Newsgroups* dataset as an example, which includes about $20,000$ emails categorized into 20 classes. It has been demonstrated that accuracy significantly improves when the header is included, compared to just using the body of the emails (Wahba et al., 2023). This improvement is likely to happen because the subject line, which is included in the header and acts as a brief summary of the email, providing crucial insights that aid in classification. The subject line contains distinct keywords that more accurately differentiate the labels. So, understanding the connection between different parts of a document and their labels is crucial for enhancing classification performance.

Moreover, other label-related factors also influence the performance of classifiers, such as class imbalance, class ambiguity, and the complexity of the boundaries separating the classes (Ho and Basu, 2002). In order to capture such aspects as well, we converted the documents from the datasets into TF-IDF vectors and, together with the associated labels, used them as inputs for task-specific meta-features extraction (Lorena et al., 2019). PsyMatrix uses a comprehensive set of meta-features as described by Lorena et al. (2019), categorized into six groups: (1) clustering, (2) complexity, (3) concept, (4) information theory, (5) general, and (6) statistical. By applying these meta-features, we can further refine our understanding and processing of classification tasks, leading to improved model performance across diverse datasets.

In this work, we used a tool called PyMFE (Alcobaça et al., 2020) to compute those meta-features from the datasets, providing additional 2825 meta-features for a dataset. For further details on this tool, and which meta-features are available, readers are encouraged to consult the official documentation.[6]

3.7 Dimensionality Reduction

Combining all the previously described processes generates a large number of meta-features for each dataset[7]. Many of them are redundant (high correlation), and certainly not all of them are important towards our objective. Hence, different dimensionality reduction techniques were combined.

Initially, we removed features that were constant or highly correlated (i.e., correlation above 0.95). Next, we applied K-means clustering to select K meta-features. Lastly, we employed variational auto-encoders (Kingma and Welling, 2013) (VAEs) to transform the remaining set of meta-features into a compact, low-dimensional representation. VAEs work by compressing data into a latent space (encoding) and then reconstructing it back to its original form (decoding). Although the reconstruction is not perfect and some information loss occurs, the technique maintains a degree of interpretability due to its reversible nature.

4 Experiments and Results

This section experimentally investigates the practical application of the PsyMatrix framework to predict the final performance Y_{ij} of a model i on a dataset j using a surrogate function $\eta_i(\phi(\psi(D_j)))$. The performance metric employed is the negative logarithm of the cross-entropy on the testing sets.

The objective is to test the ability of PsyMatrix in predicting the most suitable model for any target dataset. To do this, we evaluated the performance of various pretrained language models across a broad spectrum of text classification tasks. Specifically, we used 90% of the data for training, and the remaining 10% was employed for validation. To further enhance the reliability of our results, we employed 10-fold cross-validation, ensuring a more robust assessment of PsyMatrix's generalization ability across various datasets and reducing the impact of data variability.

[6] https://pymfe.readthedocs.io/en/latest/

[7] The total number of meta-features per dataset considered in this paper is 12205, generated by the following formula: (314 psycholinguistic features +100 topics +55 languages) ×20 summarizing functions +2825 task-specific meta-features.

Table 1: Finetuning hyper-parameters

Hyper-parameter	Value
Optimizer	AdamW
Learning rate	2.5×10^{-6}
Batch size	8
Maximum token size	1024
Maximum number of epochs	30
Early stop patience	3 epochs
16-bit (mixed) precision	Yes

4.1 Experimental Setup

In order to test PsyMatrix, we constructed a comprehensive meta-dataset repository containing performance metrics of 24 pretrained language models across 146 text classification datasets. While some datasets provide predefined train-test splits, we chose to create uniform splits for all datasets to ensure consistency. In most cases, we used $10,000$ documents for training and $2,000$ for testing. For datasets with fewer than 12,000 documents, we adjusted the number of training and testing documents proportionally, maintaining the same ratio between the two sets. More details about the datasets and pretrained models used can be found in the Appendix.

The fine-tuning process involved full model fine-tuning, where all weights of the pretrained network were updated during the fine-tuning process. This method generally yields the best results, but requires more time and resources for training. The pretrained head of base models were discarded and replaced by a new classification head, according to the number of labels of the task at hand. The fine-tuning hyperparameters employed were used as described in Table 1.

For the topic modeling feature extractor, we trained the LDA model using the training set of each dataset. The pre-processing steps were straightforward and included removing numbers, punctuation, stop words, and words with fewer than two characters. When creating the dictionary, we filtered out words that occurred fewer than 30 times or appeared in more than 50% of the documents, in order to avoid noise and overly common terms. For the hyperparameters, we set the number of topics to 100, with 10 passes and 400 iterations (following the implementation provided by Gensim[8]). Additionally, we limited the dictionary to the 10,000 most frequent words. Examples of the extracted topics can be found in the Appendix.

[8] https://pypi.org/project/gensim/

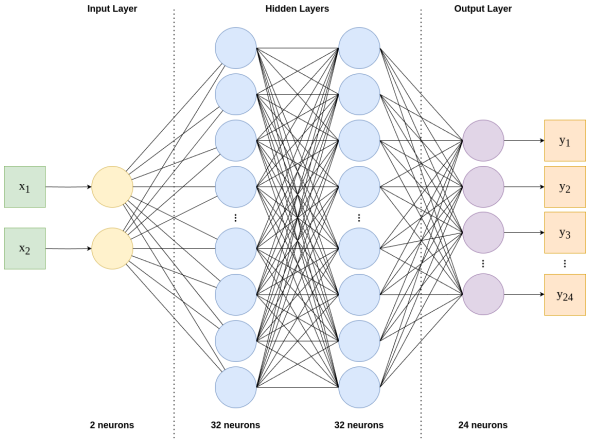


Figure 2: The neural network used as performance estimator (ranking) of a set of 24 pretrained models on a given dataset.

For the surrogate performance estimator, we utilized a multi-layer perceptron (MLP), as illustrated in Figure 2. The model's inputs consist of dataset embeddings, which were generated using a 2D Variational Autoencoder (VAE). The output of the network is the predicted validation performance for each pretrained language model used in our experiments. To properly assess the generalization ability of the MLP, we employed 10-fold cross-validation. Importantly, the data splits were made at the dataset level, meaning that some datasets were entirely excluded from the training set and used solely for validation. This ensured that during training, the model did not have access to the performance information of all datasets, allowing us to evaluate its ability to generalize to unseen datasets.

4.2 Applications

PsyMatrix can be employed in several applications related to the selection of pretrained language models across various datasets. By leveraging its ability to predict model performance and explore the relationships between datasets and model behaviors, PsyMatrix can guide users in selecting the most appropriate models for fine-tuning (optimal and near-optimal), and projecting PTM performance across the dataset embedding space. In the following, we demonstrate these capabilities through three key applications: model ranking for optimal selection, near-optimal model identification, and exploring the embedding space for performance insights.

4.2.1 Model Ranking for Optimal Model Selection

One practical application of PsyMatrix is its ability to assist in selecting the most suitable PTM for fine-tuning on a given dataset. For any dataset, we first calculate its embeddings and then use PsyMatrix to estimate the performance of a fix set of PTMs. Based on these predictions, the models can be ranked in descending order of expected performance, which can be used as a starting point for the actual fine-tuning of the PTMs, prioritizing the top models in the ranking.

To evaluate our framework prediction's performance, we compared it against two other baseline strategies. The first baseline, named *random policy*, selects models purely by chance for k trials. This simulates an unguided search, where the probability of selecting the optimal model is effectively k divided by the total number of candidate models.

The second baseline, called the *naïve policy*, uses a slightly more informed approach: it ranks the PTMs based on how frequently each one was historically the best performer. In this ranking system, the model with the highest performance across the most datasets is placed first, followed by others in decreasing order of their historical frequency of being optimal. The same ranking is then used for every new dataset. In this paper, we utilized all the information available in the meta-dataset to construct this naïve ranking.

The comparison in Figure 3 illustrates the effectiveness of the different strategies, showing the probability of identifying the optimal model within a given number of trials, denoted by k (the selection budget). The results demonstrate a clear advantage for PsyMatrix over other strategies. For example, with a budget of $k = 5$, PsyMatrix achieves success rate of $78.7 \pm 10.6\%$ in selecting the optimal model, significantly outperforming the naïve policy, which has a success rate of 36%, and the random policy, which only reaches 20%. This difference highlights the superior efficiency of PsyMatrix in guiding the model selection.

PsyMatrix's stronger performance, particularly in low-budget scenarios where resources are limited, emphasizes its practical value in real-world applications. By substantially increasing the chances of selecting the best model with fewer trials, PsyMatrix proves to be a more reliable and efficient tool for model selection compared to traditional methods.

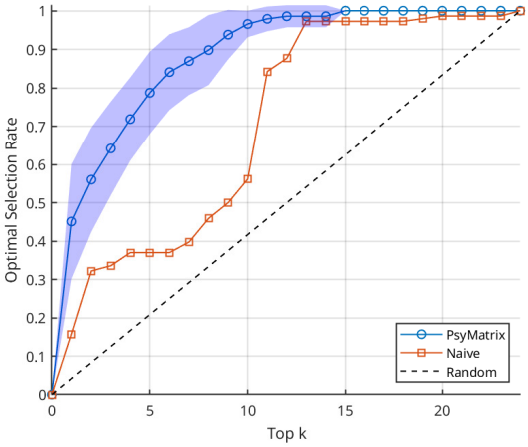


Figure 3: The probability of the optimal model being present within the top-k models of each ranking strategy. PsyMatrix consistently offers the highest probability of identifying the optimal model for any budget k.

4.2.2 Near-Optimal Model Selection

During the analysis of the experimental results performed in this paper, we observed that usually the top four or five PTMs often perform very similarly on a given dataset, achieving results close to the true optimal. This suggests that identifying models with near-optimal performance may be "good enough" for many NLP applications, particularly when budget constraints limit the ability to search exhaustively for the absolute best model.

To further demonstrate the effectiveness of our approach in this scenario, Figure 4 shows the normalized performance[9] gap between the true optimal model and the best-performing model selected among the top-k predictions. In this figure, a value of 1 corresponds to the performance of the optimal model, while 0 represents the worst-performing model.

Interestingly, when the budget is very limited ($k = 1$), the naïve policy slightly outperforms PsyMatrix, achieving 87% of the optimal model's performance compared to PsyMatrix's 83%. This demonstrates that, in extremely constrained situations, relying on the best-known model can be a reasonable strategy. However, as the budget increases ($2 \leq k \leq 12$), PsyMatrix consistently outperforms both baseline strategies. For example, with a budget of $k = 5$ trials, PsyMatrix identifies a model with an average performance that is 98% of the optimal, while the naïve policy achieves only 93%,

[9]The performance metric adopted was the negative logarithm of the cross-entropy loss on the evaluation set.

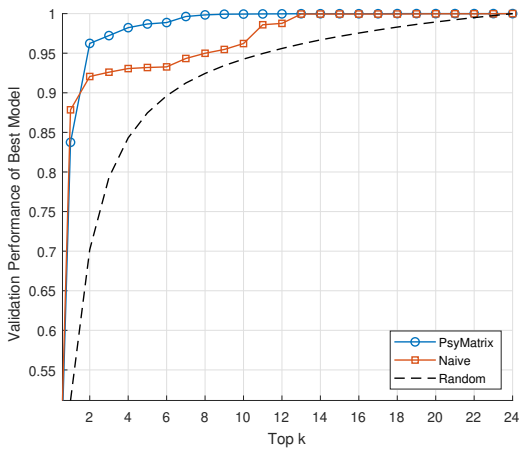


Figure 4: Performance gap between the true optimal and the best one found by each strategy, where 1.0 is the normalized performance of the optimal model, and 0.0 is the normalized performance of the worst model. PsyMatrix quickly find near-optimal solutions and is superior in all but budget $k = 1$.

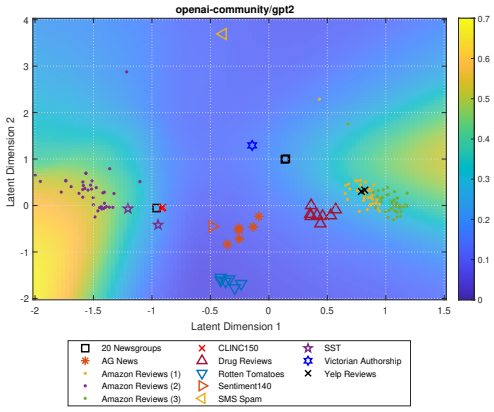
and the random policy falls further behind at 87%. This highlights that PsyMatrix is not only effective at finding the optimal model but also at selecting near-optimal models with minimal performance loss, even under moderate budget constraints.

4.2.3 Exploring the Embedding Space

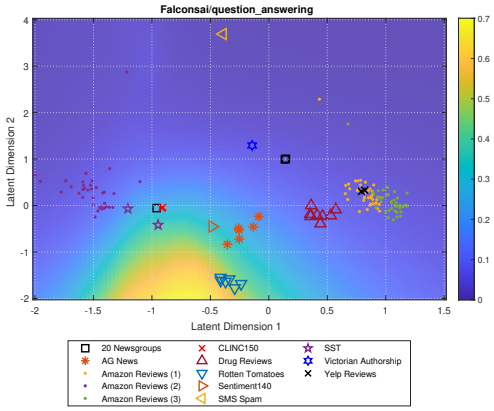
Another application of PsyMatrix is its ability to project the performance of pretrained large language models into the entire embedding space, revealing regions where the models are likely to be the optimum or near-optimum. These projections allow to visualize regions of specialization for each model, offering insights into which kinds of datasets the models are recommended for.

To achieve this, we provided the trained network of our framework a grid of embeddings, and analyzed the output from each of its output neurons, where each neuron corresponds to the estimated performance of a specific model. As an illustration, Figure 5 shows the average output for two PTMs across the validation set for each of the 10 folds: GPT-2 and FALCONS.AI Question Answering[10].

In the figure, the colors indicate the predicted probability of the given PTM being the optimal one across the embedding space. Notably, GPT-2 shows higher performance in regions associated with datasets like Amazon Reviews (along the sides of the plot), while FALCONS.AI's model demonstrates superior performance in other regions, particularly those associated with datasets such as Rotten Tomatoes and AG News (toward the bottom of the plot). By identifying these regions, is is possible to make more informed decisions for the model selection, effectively aligning specific tasks with the models that are best suited to handle them.



(a) GPT2.



(b) Fine-tuned DistilBERT for question answering.

Figure 5: Projection of PsyMatrix's score for a pretrained model being optimal across the embedding space.

This kind of information is useful for both model developers and practitioners, as it highlights where a PTM is likely to perform well and where it might face challenges. These insights can guide developers in enhancing a pretrained model by training it on more diverse datasets, or at least to help set clear boundaries regarding the types of datasets for which the model is most suitable. By understanding a model's strengths and weaknesses, we can better align it with the appropriate tasks, ultimately leading to improved performance and more reliable outcomes.

[10]This model is a fine-tuned version of DistilBERT designed for question-answering tasks.

5 Conclusion

In this paper, we introduced PsyMatrix, a novel framework for characterizing text-based datasets through psycholinguistic dimensions, topic distributions, and complexity measures. By providing interpretable and low-dimensional dataset embeddings, PsyMatrix aids in understanding the latent dataset features and accurately predicting the performance of pretrained language models finetuned for classification tasks. Our extensive validation across 146 text classification datasets demonstrated the robustness and effectiveness of PsyMatrix in guiding model selection.

In future work we will explore the application of PsyMatrix to other domains beyond text classification and expand PsyMatrix to support multiple languages.

6 Limitations

In this paper, we focused exclusively on fine-tuning performances for classification datasets. This specificity raises the first limitation: the generalization of our proposed framework to other NLP tasks has not been tested. Different tasks might exhibit unique characteristics and requirements that might not be captured by our embeddings, making it necessary to conduct further research to ascertain whether our findings hold across a broader range of applications.

Another limitation concerns the hyperparameter settings. Throughout our experiments, all finetuning processes were executed with a fixed set of training hyperparameters, such as learning rate, optimizer, maximum number of training epochs, etc. We did not undertake any optimization of these parameters for specific pretrained models or target datasets. Although this approach simplifies the experimental design, it introduces potential biases in our results. It's plausible to assume that a few finetuned models might have prematurely converged to local minima due to this lack of optimization.

So, while this paper contributes valuable insights into the interpretability and analysis of pretrained language models in context of classification tasks, these insights come with caveats that must be addressed through further, more nuanced research. Our commitment to transparency in discussing these limitations is intended to foster integrity within the research community and encourage rigorous, thoughtful examination of the framework we proposed.

References

Edesio Alcobaça, Felipe Siqueira, Adriano Rivolli, Luís P. F. Garcia, Jefferson T. Oliva, and André C. P. L. F. de Carvalho. 2020. MFE: Towards reproducible meta-feature extraction. *Journal of Machine Learning Research*, 21(111):1–5.

Esam Alzahrani, Mohammed Al Qurashi, and Leon Jololian. 2022. Comparative analysis of the use of pre-trained models to profile authors' ages and genders. In *2022 2nd International Conference on Computing and Machine Intelligence (ICMI)*, pages 1–7.

Jonathan Anderson. 1983. Lix and rix: Variations on a little-known readability index. *Journal of Reading*, 26(6):490–496.

Sebastian Arango, Fabio Ferreira, Arlind Kadra, Frank Hutter, and Josif Grabocka. 2024. Quick-tune: Quickly learning which pretrained model to finetune and how. In *International Conference on Learrning Representations, ICLR'24*.

R. H. Baayen, R. Piepenbrock, and L. Gulikers. 1995. *CELEX2 LDC96L14*. Linguistic Data Consortium, Philadelphia.

Shivam Bansal and Chaitanya Aggarwal. 2024. Textstat: Python package to help in text statistics. Original software developed by Shivam Bansal and contributors.

Santosh Kumar Barnwal and Uma Shanker Tiwary. 2017. Using psycholinguistic features for the classification of comprehenders from summary speech transcripts. In *Intelligent Human Computer Interaction*, pages 122–136, Cham. Springer International Publishing.

Hilan Bensusan, Christophe Giraud-Carrier, and Claire Kennedy. 2000. A higher-order approach to meta-learning. In *Proceedings of the ECML'2000 workshop on Meta-Learning: Building Automatic Advice Strategies for Model Selection and Method Combination*, pages 109 – 117. ECML'2000.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Daniel Bolya, Rohit Mittapalli, and Judy Hoffman. 2021. Scalable diverse model selection for accessible transfer learning. In *Advances in Neural Information Processing Systems*.

Pavel Brazdil, Jan N. van Rijn, Carlos Soares, and Joaquin Vanschoren. 2022. *Dataset Characteristics (Metafeatures)*, pages 53–75. Springer International Publishing, Cham.

Scott A. Crossley, Kristopher Kyle, and Danielle S. McNamara. 2016. The tool for the automatic analysis of text cohesion (TAACO): Automatic assessment of local, global, and text cohesion. *Behavior Research Methods*, 48(4):1227–1237.

Ali Daban, Siti Armiza Mohd Aris, and Mohd Syahid Mohd Anuar. 2023. Ferrous metal classifications based on sparks pattern using CNN pretrained models. In *2023 IEEE Symposium on Computers & Informatics (ISCI)*, pages 31–35.

Harrison Edwards and Amos Storkey. 2017. Towards a neural statistician. In *International conference on learning representations, ICLR'17*.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Arthur C. Graesser, Danielle S. McNamara, and Jonna M. Kulikowich. 2011. Coh-Metrix: Providing multilevel analyses of text characteristics. *Educational Researcher*, 40(5):223–234.

Arthur C. Graesser, Danielle S. McNamara, Max M. Louwerse, and Zhiqiang Cai. 2004. Coh-Metrix: Analysis of text on cohesion and language. *Behavior Research Methods, Instruments, & Computers*, 36(2):193–202.

Robert Gunning. 1952. *The Technique of Clear Writing*. McGraw-Hill, New York.

Tin Kam Ho and M. Basu. 2002. Complexity measures of supervised classification problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):289–300.

Hadi S. Jomaa, Lars Schmidt-Thieme, and Josif Grabocka. 2021. Dataset2vec: learning dataset meta-features. *Data Mining and Knowledge Discovery*, 35:964–985.

Diederik P. Kingma and Max Welling. 2013. Auto-encoding variational bayes. *CoRR*, abs/1312.6114.

Hao Li, Charless C. Fowlkes, Han Yang, Onkar Dabeer, Zhuowen Tu, and Stefan Soatto. 2023. Guided recommendation for model fine-tuning. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3633–3642.

Guido Lindner and Rudi Studer. 1999. AST: Support for algorithm selection with a CBR approach. In *European Conference on Principles of Data Mining and Knowledge Discovery*.

Ana C. Lorena, Luís P. F. Garcia, Jens Lehmann, Marcilio C. P. Souto, and Tin Kam Ho. 2019. How complex is your classification problem? A survey on measuring classification complexity. *ACM Comput. Surv.*, 52(5).

Vincent Quirante Malic, Anamika Kumari, and Xiaozhong Liu. 2023. Racial skew in fine-tuned legal AI language models. In *2023 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 245–252.

Danielle S. McNamara, Arthur C. Graesser, Philip M. McCarthy, and Zhiqiang Cai. 2014. *Automated Evaluation of Text and Discourse with Coh-Metrix*. Cambridge University Press.

Yonghong Peng, Peter A. Flach, Carlos Soares, and Pavel Brazdil. 2002. Improved dataset characterisation for meta-learning. In *Discovery Science*, pages 141–152, Berlin, Heidelberg. Springer Berlin Heidelberg.

Chunping Qiu, He Li, Wenyue Guo, Xin Chen, Anzhu Yu, Xiaochong Tong, and Michael Schmitt. 2022. Transferring transformer-based models for cross-area building extraction from remote sensing images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15:4104–4116.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training. Technical report, OpenAI. Preprint.

Fei Ren, Dong Lu, and Naxin Cui. 2023. A deep transfer learning framework for Li-ion battery temperature prediction based on LSTM pretraining model. In *2023 IEEE 2nd International Power Electronics and Application Symposium (PEAS)*, pages 1170–1175.

Adriano Rivolli, Luís P.F. Garcia, Carlos Soares, Joaquin Vanschoren, and André C. P. L. F. de Carvalho. 2022. Meta-features for meta-learning. *Knowledge-Based Systems*, 240:108101.

Dmitri Roussinov and Serge Sharoff. 2023. BERT goes off-topic: Investigating the domain transfer challenge using genre classification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 468–483, Singapore. Association for Computational Linguistics.

Cullen Schaffer. 1994. Cross-validation, stacking and bi-level stacking: Meta-methods for classification learning. In *Selecting Models from Data*, pages 51–59, New York, NY. Springer New York.

Elena Shushkevich, Mikhail Alexandrov, and John Cardiff. 2022. Bert-based classifiers for fake news detection on short and long texts with noisy data: A comparative analysis. In *Text, Speech, and Dialogue*, pages 263–274, Cham. Springer International Publishing.

Daniel Simig, Tianlu Wang, Verna Dankers, Peter Henderson, Khuyagbaatar Batsuren, Dieuwke Hupkes, and Mona Diab. 2022. Text characterization toolkit (TCT). In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 72–87, Taipei, Taiwan. Association for Computational Linguistics.

So Young Sohn. 1999. Meta analysis of classification algorithms for pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1137–1144.

Zhiqiang Tang, Haoyang Fang, Su Zhou, Taojiannan Yang, Zihan Zhong, Tony Hu, Katrin Kirchhoff, and George Karypis. 2024. Autogluon-multimodal (automm): Supercharging multimodal automl with foundation models. *Preprint*, arXiv:2404.16233.

M. C. Templin. 1957. *Certain Language Skills in Children: Their Development and Interrelationships*, volume 10. University of Minnesota Press, Minneapolis, MN.

Yasmen Wahba, Nazim Madhavji, and John Steinbacher. 2023. A comparison of SVM against pre-trained language models (PLMs) for text classification tasks. In *Machine Learning, Optimization, and Data Science*, pages 304–313, Cham. Springer Nature Switzerland.

David Wolpert. 1996. The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8.

Shaomin Zheng and Meng Yang. 2019. A new method of improving bert for text classification. In *Intelligence Science and Big Data Engineering. Big Data and Machine Learning*, pages 442–452, Cham. Springer International Publishing.

Appendix

A Pretrained Language Models

In this study, we evaluated the performance of various pretrained language models (PTMs), which are available through the Hugging Face model hub. These models span a range of architectures and capabilities, and were selected to provide a comprehensive comparison across different natural language processing tasks. Below is the list of pretrained models (referenced by their Hugging Face repository IDs) used in our experiments:

1. openai-community/gpt2
2. google-bert/bert-base-multilingual-cased
3. google-bert/bert-base-cased
4. FacebookAI/roberta-base
5. FacebookAI/xlm-roberta-base
6. albert/albert-base-v2
7. xlnet/xlnet-base-cased
8. microsoft/mpnet-base
9. google/fnet-base
10. allenai/longformer-base-4096
11. studio-ousia/luke-base
12. studio-ousia/luke-japanese-base
13. bigscience/bloom-560m
14. bigscience/bloomz-560m
15. funnel-transformer/medium-base
16. Falconsai/question_answering
17. deepmind/language-perceiver
18. kssteven/ibert-roberta-base
19. uw-madison/nystromformer-1024
20. uw-madison/yoso-4096
21. flaubert/flaubert_base_cased
22. nghuyong/ernie-3.0-base-zh
23. facebook/opt-125m
24. facebook/opt-1.3b

B Datasets

All experiments in this work focused on text classification tasks. We curated a total of 146 distinct datasets, derived from 11 base datasets. These base datasets are widely used in natural language processing (NLP) tasks and cover a range of text classification challenges, such as sentiment analysis, topic classification, and intent detection. The datasets used are listed below:

1. 20 Newsgroups
2. AG News
3. Amazon Reviews
4. CLINC 150
5. Drug Reviews (Drugs.com)
6. Rotten Tomates
7. Sentiment 140
8. SMS Span
9. Stanford Sentiment Treebank
10. Victorian Authorship
11. Yelp Reviews

To create the 146 distinct datasets, we generated subsets by varying the input features and target labels available within each base dataset. This approach allowed us to explore different aspects of each dataset for more comprehensive experimentation. For example:

- **20 Newsgroups**: this dataset was divided into three subsets based on different input features. One subset used only the email body as input, another used only the subject line, and a third used both the subject and body combined.
- **Sentiment Analysis**: for datasets involving sentiment analysis, such as Amazon Reviews and Rotten Tomatoes, we created multiple subsets, including both binary classification (positive/negative sentiment) and balanced versions to ensure equal representation of classes.
- **Amazon Reviews**: we further divided the dataset by its product categories, creating separate subsets for each sub-category to assess model performance on different domains.

The Amazon Reviews dataset, due to its extensive size and multiple categories, contributed significantly to the total number of subsets. To make the

analysis more manageable, we grouped the Amazon Reviews subsets into three clusters for visualization in the figures, as described in Table 2.

Table 2: Amazon Reviews Sub-groups

Dataset	Inputs	# Classes
Amazon Review (1)	summary, text	5
	summary	5
Amazon Review (2)	summary, text	2
Amazon Review (3)	summary	2

This process of creating multiple subsets allowed us to examine PsyMatrix's performance on a wide variety of tasks, ensuring robust evaluation across different types of text classification problems. For reproducibility, all datasets used in this study have been made publicly available on Hugging Face Hub.[11]

C Summarizing Functions

This section details the summarizing functions chosen to extract meta-features from the dataset's basic features. Each function was selected based on its ability to provide a comprehensive statistical overview of the data, facilitating deeper insights into its underlying distribution, spread, central tendency, and variability:

1. Pearson product-moment correlation coefficients (corrcoef)
2. Interquartile range (iqr)
3. Unbiased estimator of the variance of the k-statistic (kstatvar)
4. Fisher's coefficient of kurtosis (kurtosis)
5. Maximum (max)
6. Arithmetic mean (mean)
7. Median absolute deviation (mad)
8. Minimum (min)
9. Modal (most common) value (mode)
10. Modal occurrences (mode_count)
11. First moment about the mean (moment)
12. Range of values (ptp)
13. First quartile (q1)
14. Second quartile (q2)
15. Third quartile (q3)
16. Standard error of the mean (sem)
17. Skewness (skew)
18. Standard deviation (std)
19. Variance (var)
20. Coefficient of variation (variation)

[11] https://huggingface.co/PsyMatrix

D Ablation Studies: VAE

In order to determine the importance of Variational Autoencoders (VAE) in our framework, in this section we present an ablation study. Figure 6 illustrates the impact of removing the VAE-based feature compression on our model's prediction performance. The performance metrics clearly demonstrate a substantial degradation when VAE is omitted. This drop in performance highlights the VAE's critical role in retaining meaningful information in a compressed representation, which is essential for maintaining the accuracy and robustness of our predictive models.

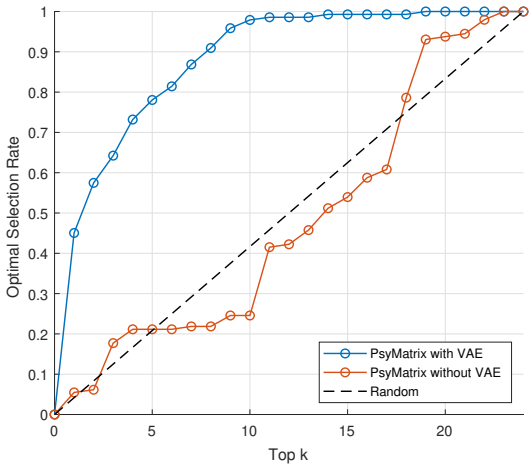


Figure 6: Performance comparison when removing the feature compressions provided by VAE. This comparison illustrates the significant contribution of VAE to maintaining high predictive accuracy in our models.

These results confirm that effective feature compression is important for our framework, hence supporting our decision to integrate VAE into our dataset embedding pipeline, ensuring that PsyMatrix can handle high-dimensional data efficiently.

E Topic Modeling

In this section, we provide examples of topics extracted from the trained corpus employed in this paper. The objective is to offer readers an overview of the types of topics identified by the Latent Dirichlet Allocation (LDA) technique. We encourage readers to follow the instructions in our framework to download and explore the trained LDA model for a deeper analysis and understanding.

Table 3 displays a selection of sample topics extracted using LDA. Each entry includes a list of the top-10 words associated with a specific topic,

accompanied by their respective probability scores. These scores quantify the likelihood of each word's occurrence within the topic, thus highlighting its importance within the cluster. For instance, Topic 1 is characterized by words such as light, power, heart, God, and truth, suggesting themes of spiritual or existential nature.

It is important to clarify that we did not perform hyper-parameter optimization to enhance topic coherence or diversity. Nevertheless, the topics extracted appear to be meaningful and useful for analyzing the datasets.

Table 3: Top-10 words for some extracted topics

Topic 1		Topic 5		Topic 8		Topic 27		Topic 33	
light	0.14	guitar	0.23	use	0.44	needed	0.13	gained	0.07
power	0.11	author	0.12	recommend	0.20	bleeding	0.10	appetite	0.05
charge	0.06	page	0.06	working	0.09	symptoms	0.09	fiction	0.05
heart	0.06	lose	0.04	highly	0.08	headaches	0.06	instructions	0.03
god	0.06	improvement	0.03	recommended	0.05	hair	0.05	yeast	0.03
blue	0.03	acoustic	0.02	buying	0.04	red	0.05	cough	0.03
saved	0.02	generic	0.02	uncomfortable	0.01	clean	0.05	pros	0.03
truth	0.02	controls	0.02	likes	0.01	feet	0.04	tuning	0.03
present	0.02	electric	0.02	rated	0.00	send	0.04	woke	0.02
attack	0.02	led	0.02	juice	0.00	straight	0.04	cons	0.02
Topic 37		Topic 39		Topic 66		Topic 72		Topic 78	
pages	0.05	sound	0.23	nice	0.26	phone	0.42	free	0.16
major	0.05	music	0.09	movie	0.14	sex	0.07	daughter	0.08
history	0.05	play	0.08	enjoy	0.08	periods	0.06	husband	0.08
pictures	0.04	acne	0.08	boring	0.06	calls	0.03	women	0.08
war	0.04	player	0.05	liked	0.06	driving	0.02	woman	0.07
text	0.03	hear	0.05	enjoyed	0.05	jane	0.01	ring	0.06
middle	0.03	voice	0.03	movies	0.04	reaction	0.01	men	0.06
caused	0.03	heard	0.03	watching	0.03	sister	0.01	update	0.05
mobile	0.02	pedal	0.02	horror	0.02	breast	0.01	satisfied	0.04
state	0.02	fan	0.02	plain	0.02	inch	0.01	finds	0.02