AutoML Meets Hugging Face: Domain-Aware Pretrained Model Selection for Text Classification

Parisa Safikhani[1,2] **David Broneske**[1]
[1]The German Centre for Higher Education Research and Science Studies (DZHW), Germany
[2]University of Magdeburg, Germany
safikhani@dzhw.eu, broneske@dzhw.eu

Abstract

The effectiveness of embedding methods is crucial for optimizing text classification performance in Automated Machine Learning (AutoML). However, selecting the most suitable pre-trained model for a given task remains challenging. This study introduces a comprehensive corpus of pre-fine-tuned models from the Hugging Face Model Hub, annotated with domains and dataset descriptions, to enhance text classification tasks. By leveraging this corpus, we evaluated the integration of pre-fine-tuned models into AutoML systems, demonstrating substantial performance gains across various datasets compared to baseline methods. Despite some inaccuracies in domain recognition, the results underscore the corpus' potential to streamline model selection and reduce computational costs.

1 Introduction

The advent of large language models (LLMs) has significantly advanced natural language processing (NLP), offering powerful tools for tasks such as text classification, summarization, and translation (Devlin et al., 2018). Fine-tuning these models for specific tasks has traditionally been the standard approach to achieving optimal performance. However, fine-tuning is resource-intensive, requiring substantial computational power and time, which may not be feasible for all practitioners (Wolf et al., 2020).

Simultaneously, AutoML automates tasks like feature and model selection, offering a streamlined approach to machine learning (He et al., 2021). Integrating LLMs into AutoML can boost NLP performance by leveraging their rich linguistic representations (Tornede et al., 2023).

A practical alternative to fine-tuning is utilizing pre-fine-tuned LLMs available in repositories such as Hugging Face. These models have been trained on specific tasks or domains and offer ready-to-use LLMs that can be incorporated into AutoML classifiers. This approach can improve performance while mitigating the resource constraints associated with fine-tuning.

Despite their potential, pre-fine-tuned LLMs from repositories like Hugging Face remain underexplored as text representation methods in AutoML. This study bridges this gap by developing an interface to a domain-annotated corpus of pre-fine-tuned models and evaluating their impact on classification performance across seven diverse text classification tasks.

This study enhances AutoML-based text classification by introducing a structured corpus of pre-fine-tuned models annotated with domain-specific metadata to optimize model selection. By systematically mapping models to tasks based on domain alignment, we demonstrate substantial performance gains while reducing computational overhead. The findings highlight a scalable and resource-efficient approach for integrating pre-trained representations into AutoML frameworks, making advanced NLP capabilities more accessible.

2 Related Works

LLMs and Contextual Embeddings: Contextual embeddings from fine-tuned LLMs outperform static methods like TF-IDF and Word2Vec in classification tasks by creating highly separable vector spaces (Pietro, 2020; Koroteev, 2021; Andrade, 2023; Safikhani and Broneske, 2023a). While fine-tuned LLMs achieve superior results, their computational cost limits their applicability. Pre-fine-tuned models, tailored for specific tasks, provide a scalable alternative (Wolf et al., 2020).

Text Representations in AutoML: AutoML frameworks like Auto-PyTorch aim to automate feature extraction, model selection, and hyperparameter tuning (Zimmer et al., 2021; Feurer et al.,

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 4: Student Research Workshop), pages 466–473

2015). Despite this, they often rely on basic text representations like one-hot encoding. Recent research highlights the benefits of integrating advanced embeddings into AutoML systems. For instance, Safikhani and Broneske (2023b) demonstrated the effectiveness of fine-tuned BERT embeddings for binary classification in Auto-PyTorch. However, leveraging pre-fine-tuned LLMs for AutoML remains underexplored.

Open-Source Pre-Fine-Tuned Models: The Hugging Face Model Hub offers many pre-fine-tuned models optimized for tasks such as text classification, sentiment analysis, and named entity recognition (Wolf et al., 2020). These models (see for instance `BioBERT` (Lee et al., 2020), `SciBERT` (Beltagy et al., 2019), and `XLM-R` (Conneau, 2019) address domain-specific needs and reduce reliance on fine-tuning. Comprehensive model cards (Mitchell et al., 2019) provide transparency, aiding in model selection and reproducibility.

While pre-fine-tuned LLMs show promising results, their integration into AutoML classifiers has not been systematically studied. This research addresses this gap by evaluating the impact of pre-fine-tuned models as text representation methods in AutoML, focusing on their performance across diverse text classification tasks.

3 Methodology

In order to achieve our two goals of *interfacing* and *selecting* pre-fine-tuned models, we implement the following two phases.

3.1 Pre-trained Model Repository Integration

In the first phase of our methodology, we established an interface between a model repository (Hugging Face) and our AutoML framework (Auto-PyTorch). This integration enables the AutoML system to leverage a rich corpus of pre-trained NLP models, facilitating model reuse for downstream text classification tasks. Retrieving pre-trained (and fine-tuned) models from repositories like Hugging Face is critical for enhancing AutoML, as it allows rapid deployment and adaptation to new tasks without the high computational cost of training models from scratch. We implemented a configurable interface to the Hugging Face Hub API[1] that allows Auto-PyTorch to programmatically query and retrieve models. This retrieval process provided a diverse pool of candidate models, each trained on various text classification datasets and tasks. However, many models on the repository lacked clear documentation of their intended domains. To address this, we analyzed the datasets used for each model's fine-tuning as a proxy for its domain, using those dataset references to infer the types of tasks or domains for which each model is best suited.

3.2 Selection of Domain-Specific Models

Given the possibility of retrieving the pre-trained models from Hugging Face, the next phase implements the selection of a specific model. Hence, the domain of the models needs to be matched with the domain of the datasets.

3.2.1 Domain Definition from Literature

We conducted a literature review to identify key domains in text classification, as shown in Table 1. These domains, supported by foundational references, provide a framework for contextualizing models and analyzing domain representation in the corpus. We curated a list of 30 domains from existing literature (e.g., Sentiment Analysis, Spam Detection, Hate Speech Detection).

3.2.2 Domain Identification of Hugging Face Models

To identify the domain of models retrieved from the Hugging Face API, we mapped model names to dataset descriptions when explicit model descriptions were not available in the metadata.

As the collected domain labels, such as "Hate Speech Detection," often lack sufficient contextual richness and may overlook intricate nuances, we employed ChatGPT to generate extended descriptions. This approach bridges the semantic gap between concise labels and detailed model documentation, enhancing matching precision by capturing variations in terminology used across different contexts.

To map these models to a domain, we compared the model's description against the generated domain descriptions using sentence embeddings from `all-MiniLM-L6-v2` provided by Sentence-BERT (Reimers, 2019). We applied cosine similarity (Singhal et al., 2001) between the embeddings to assign the most semantically relevant domain to each model.

We selected a pre-fine-tuned model from the Hugging Face repository for each evaluation dataset based on the recognized domain. A fall-

[1] https://huggingface.co/docs/hub/api

Domain	Generated Description by ChatGPT
Emotion Cause Extraction (Ghazi et al., 2015)	Identifying the reasons or triggers for specific emotions in text.
Social Media Behavior Analysis (Aral and Walker, 2012)	Analyzing user behavior on social media platforms.
Rhetorical Structure Classification (Mann and Thompson, 1988)	Classifying rhetorical structures in discourse.
Spam Detection (Guzella and Caminhas, 2009)	Classifying emails or messages as spam or legitimate.
Language Identification (Jauhiainen et al., 2019)	Detecting the language of text, especially in multilingual settings.
Sentiment Analysis (Pang et al., 2008)	Detecting opinions, emotions, and sentiments in text.
Topic Classification (Blei et al., 2003)	Assigning topics or categories to text documents.
Emotion Recognition (Cowie et al., 2001)	Identifying emotions such as joy, sadness, anger, and fear in text.
Intent Classification (Liu et al., 2019)	Understanding the purpose or intent behind user queries.
Hate Speech Detection (Davidson et al., 2017)	Detecting hate speech, toxic, or abusive language in the text.
Textual Entailment (Bowman et al., 2015)	Determining if one text logically follows from another.
Document Classification (Rios and Kavuluru, 2018)	Categorizing entire documents into predefined classes.
Fake News Detection (Shu et al., 2017)	Detecting false or misleading news articles.
Aspect-Based Sentiment Analysis (Pontiki et al., 2016)	Analyzing sentiment specific to different aspects of a product or service.
Sarcasm Detection (Joshi et al., 2017)	Identifying sarcasm or ironic statements in the text.
Propaganda Detection (Da San Martino et al., 2019)	Detecting manipulative or biased content in text.
Irony Detection (Van Hee et al., 2018)	Identifying ironic statements in the text.
Argument Mining (Van Hee et al., 2018)	Analyzing arguments and their structures in the text.
Deception Detection (Fitzpatrick et al., 2015)	Detecting lies, fraud, or deceptive statements in text.
Lexical Complexity Prediction (Shardlow, 2013)	Predicting the complexity or difficulty of words in the text.
Politeness Classification (Danescu-Niculescu-Mizil et al., 2013)	Classifying text based on politeness levels.
Coreference Resolution (Lee et al., 2017)	Linking pronouns and entities to their references.
Genre Classification (Stamatatos et al., 2000)	Classifying text into genres such as fiction, non-fiction, etc.
Temporal Information Extraction (Bethard, 2013)	Extracting time-related information from text.
Claim Verification and Fact-Checking (Thorne and Vlachos, 2018)	Verifying the truth of claims in text.
Persuasiveness Classification (Habernal and Gurevych, 2016)	Classifying how persuasive text is.
Privacy Risk Classification (Biega et al., 2020)	Detecting privacy risks in text data.
Media Bias Detection (Baly et al., 2020)	Identifying bias in news or media content.
Speech Emotion Classification (Busso et al., 2013)	Recognizing emotions from spoken text or transcripts.
Multimodal Text Classification (Kiela et al., 2019)	Classifying text combined with other modalities like images or audio.

Table 1: Categorized Domains in Text Classification with Descriptions Generated Using ChatGPT and Foundational References, Serving as a Framework for Similarity-Based Domain Assignments.

back model (`all-MiniLM-L6-v2`) was used if no specific model was available for the recognized domain. Sentence embeddings for the datasets were then generated using the selected model.

Furthermore, it supports multi-task scenarios, making it a versatile choice when domain-specific models are unavailable.

3.3 Domain Identification of a given Datasets

To assign domains to our evaluation datasets, we implemented a comprehensive zero-shot classification approach using the cross-encoder/nli-deberta-v3-small model, particularly suited for its ability to interpret and classify complex data directly. This method is preferred over cosine similarity because it allows for a more dynamic interpretation of text semantics rather than just vector alignment, which is critical in understanding the nuanced thematic content of datasets that might not be immediately apparent through traditional vector space models.

Our process begins by selecting a representative subset of text samples from each class within the dataset to ensure comprehensive coverage of all potential categories within the classification task. These samples are systematically evaluated against our predefined domain names using the zero-shot model, which assesses the likelihood of each text sample fitting into each potential domain. Zero-shot learning is particularly effective because it evaluates the semantic content of the samples in a contextual manner, thus allowing for accurate classifications based on the inherent meanings and not merely on the superficial similarity of words or phrases.

To ensure robust domain assignment, we compute similarity scores between each text sample and each domain, then calculate the average similarity score across all classes for each domain. This averaging is crucial as it ensures that the domain assignment reflects the diversity of the entire dataset and is not biased toward dominant themes within any single class. Finally, the domain with the highest average similarity score is assigned to the dataset. This method is superior to cosine similarity as it provides a balanced and accurate domain assignment that effectively captures the complexity and diversity of the dataset. It utilizes the strengths of zero-shot learning to adapt to new and unseen categories seamlessly, making it more adaptable to datasets with varied and evolving themes.

4 Experiment

The experimental workflow evaluated the utility of pre-fine-tuned language models from Hugging Face for diverse text classification tasks. The process involved multiple steps, including collecting

model metadata, domain recognition, dataset preparation, model selection, and evaluation. Below, we detail each step of the experimental setup.

4.1 Dataset Preparation

To evaluate the models, we used datasets from Kaggle[2], including Colbert (humor), IMDB Reviews (sentiment analysis), Cyberbullying Comments, Disaster Tweets Detection, Emotion Detection from Text, Amazon Reviews, and an annotated dataset for framing detection (Avetisyan and Broneske, 2021) to prevent data snooping. More detailed information about these datasets is provided in table 2.

4.2 Experimental Setup

The generated embeddings were split into training and testing sets (80/20 split) and used to train classification models. Auto-PyTorch was utilized to automatically configure and optimize the classification pipeline, employing a k-fold cross-validation strategy for robust evaluation.

We assessed model performance primarily using metrics tailored for imbalanced datasets. AUPRC was used for binary classification tasks to evaluate precision-recall trade-offs effectively, and micro F1-Score was employed for robust evaluation in multi-class settings.

The experiments were conducted on a high-performance system featuring an NVIDIA A100 GPU with 40 GB VRAM, dual Intel Xeon Gold 5220R CPUs, and 376 GB RAM, running Ubuntu 20.04 LTS. Key software included Python 3.8, PyTorch 1.9, Hugging Face Transformers 4.9, and Auto-PyTorch 0.0.6, optimized for efficient model training and inference.

5 Results and Discussion

The results of our evaluation, presented in Table 3, highlight the effectiveness of the proposed **Corpus-Driven Domain Mapping (CDDM)** approach, which utilizes pre-fine-tuned models as text representation methods for Auto-PyTorch. The performance of models selected from the constructed corpus was compared against the baseline Auto-PyTorch classifier, which uses one-hot encoding as the default text representation method. These comparisons were conducted across seven text classification datasets to evaluate the impact of domain-specific pre-trained representations.

[2]https://www.kaggle.com/

Performance Overview

The evaluation results show that integrating pre-fine-tuned models into Auto-PyTorch improves performance on various text classification datasets. This effectiveness depends on domain recognition accuracy, which affects model alignment with specific tasks. Below, we present key outcomes by recognized domains and corresponding pre-fine-tuned models from the Hugging Face repository:

Media Bias Detection: This model showed substantial performance improvements across several datasets. On the `Colbert` dataset, designed for humor detection but misclassified as media bias, the model achieved an AUPRC of 92.3% compared to the baseline of 52%. Similarly, on the `Cyberbullying Comments` dataset, where the domain was correctly identified as media bias, the model attained an AUPRC of 70.2%, outperforming the baseline of 46.55%. These results highlight the robustness of pre-fine-tuned models, even when domain recognition is not entirely accurate. However, precise domain alignment remains crucial for unlocking the full potential of the corpus.

Sexism and Misogyny Detection: On the `Disaster Tweets Detection` dataset, the domain recognition step correctly assigned sexism and misogyny detection. This resulted in a significant performance boost, with an AUPRC of 44.7% compared to 19.01%. Accurate domain recognition was instrumental in leveraging the model effectively for this task.

User Stance Classification: For the `IMDB Reviews` dataset, the recognized domain of stance classification was a reasonable match given the sentiment-related nature of the task. The model achieved an AUPRC of 67.5%, surpassing the baseline of 50.63%. This suggests that while the selected model performed well, assigning a sentiment-specific model could yield even better results.

Emotion Recognition: On the `Emotion Detection from Text` dataset, the domain recognition was accurate, resulting in an AUPRC of 71.7%, significantly higher than the baseline of 51.66%. This highlights the value of precise domain matching in maximizing the corpus's utility.

Genre Classification: On the `Amazon Reviews` dataset, the domain was correctly identified as

Dataset Binary	Number of Texts	Number of Classes	Average Text Length	Balanced
ColBERT	200,000	2 (Formal, Informal)	20 words	Yes
Disaster Tweets Detection	11,223	2 (Disaster, Not)	30 words	No
Cyberbullying Comments	115,661	2 (Cyberbullying, Not)	12 words	Yes
Framing Detection	4,063	2 (Framed, Not Framed)	25 words	No
IMDB Reviews	50,000	2 (Positive, Negative)	230 words	Yes
Dataset Multi-class	**Number of Texts**	**Number of Classes**	**Average Text Length**	**Balanced**
Amazon Reviews	17,337	3	33 words	No
Emotion Detection from Text	40,000	13	14 words	Yes

Table 2: Overview of Datasets for Binary and Multi-class Text Classification Tasks

Dataset Binary	Baseline (AUPRC %)	CDDM (AUPRC %)	Recognized Domain
ColBERT	52	92.3	Media Bias Detection
Disaster Tweets Detection	19.01	44.7	Sexism and Misogyny Detection
Cyberbullying Comments	52.00	92.3	Media Bias Detection
Framing Detection	46.55	70.2	Media Bias Detection
IMDB Reviews	50.63	67.5	User Stance Classification in Online Debates
Dataset Multi-class	**Baseline (Micro F1 %)**	**CDDM (Micro F1 %)**	**Recognized Domain**
Amazon Reviews	48.46	80.7	Genre Classification
Emotion Detection from Text	51.66	71.07	Propaganda Detection

Table 3: Performance Comparison of Pre-Fine-tuned Models Selected via Corpus-Driven Domain Mapping (CDDM) and Baseline Representations Across Text Classification Tasks

genre classification. The model achieved an impressive AUPRC of 80.7%, emphasizing the advantages of accurate domain recognition and the potential of the Hugging Face corpus for domain-specific tasks.

Propaganda Detection: On the `Framing Detection` dataset, the recognized domain was media bias detection rather than propaganda detection. Despite this misalignment, the model achieved an AUPRC of 70.2%, outperforming the baseline of 46.55%. This result underscores the need for more accurate domain recognition to fully utilize the potential of the corpus.

The corpus of pre-fine-tuned models from Hugging Face, annotated with domains and dataset descriptions, represents a valuable resource for advancing text classification tasks. Its diversity and systematic structure streamline model selection, reducing the need for extensive fine-tuning and saving computational resources.

The experiments demonstrate the utility of this corpus, with substantial performance gains over baseline models, even when domain recognition was occasionally imprecise. The corpus addresses a critical gap in NLP workflows by mapping datasets to suitable models based on domain alignment.

This study shows that the corpus offers a scalable framework for integrating pre-tuned models in AutoML systems like Auto-PyTorch. Allowing task-specific model selection and optimization has proven effective in improving performance across various text classification tasks. The results emphasize that accurate domain recognition significantly boosts performance, indicating the potential for greater efficiency and wider application in NLP workflows with further refinements.

In summary, the Hugging Face corpus compiled in this study is not just a collection of models but an indispensable resource that has already demonstrated its impact through improved text classification performance. With further refinement, particularly in domain recognition and model alignment, this corpus can potentially set a new standard for leveraging open-source models in diverse and complex NLP tasks within AutoML frameworks.

6 Conclusion and Future Works

This study introduced a corpus of pre-fine-tuned models from Hugging Face enriched with domain annotations and dataset descriptions, demonstrating its utility for enhancing text classification tasks. The experimental results highlight how this resource improves model performance and streamlines integration into automated pipelines, reducing the need for fine-tuning.

In conclusion, the Hugging Face corpus represents a critical step toward scalable and efficient NLP solutions. Refinements in domain recognition and alignment hold the potential to revolutionize the use of pre-fine-tuned models in AutoML, advancing text classification and broader NLP tasks.

Future work will focus on improving domain recognition accuracy through advanced methods such as supervised learning or knowledge graph-based approaches. Additionally, it will evaluate the corpus with a more diverse range of datasets, including low-resource languages.

Future work will optimize text representation methods for specific datasets to enhance the proposed corpus's utility in AutoML systems. We will develop a multi-model evaluation framework that aligns three semantically similar pre-fine-tuned models from the corpus to each dataset based on domain similarity scores and zero-shot classification results. These models will be assessed using AutoML techniques supported by Auto-PyTorch, enabling efficient performance evaluation through automated hyperparameter optimization and model selection. By employing multi-fidelity optimization methods like Successive Halving and Hyperband, we aim to identify the most effective model early in training, reducing computational costs. This method balances model performance with efficiency while preserving the domain-specific strengths of our corpus.

Limitations

While the proposed corpus demonstrates significant potential, several limitations should be noted.

First, the evaluation datasets, though diverse, are not comprehensive and do not fully capture the complexity of real-world text classification tasks.

Second, while domain recognition methods are effective, they have accuracy limitations. For instance, the Colbert dataset, designed for humor detection, was misclassified as media bias, highlighting the need for more nuanced approaches like supervised learning or knowledge graph-based mapping.

Despite these challenges, the results highlight the potential of the Hugging Face corpus as a valuable resource for text classification and other NLP tasks, with opportunities for further refinement to enhance its utility in the AutoML domain.

References

Claudio MV de et al. Andrade. 2023. On the class separability of contextual embeddings representations – or "the classifier does not matter when the (text) representation is so good!". *Information Processing and Management*, 60:103336.

Sinan Aral and Dylan Walker. 2012. Identifying influential and susceptible members of social networks. *Science*, 337(6092):337–341.

Hayastan Avetisyan and David Broneske. 2021. Identifying and understanding game-framing in online news: Bert and fine-grained linguistic features. In *Proceedings of the 4th International Conference on Natural Language and Speech Processing (ICNLSP 2021)*, pages 95–107.

Ramy Baly, Giovanni Da San Martino, James Glass, and Preslav Nakov. 2020. We can detect your bias: Predicting the political ideology of news articles. *arXiv preprint arXiv:2010.05338*.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*.

Steven Bethard. 2013. Cleartk-timeml: A minimalist approach to tempeval 2013. In *Second joint conference on lexical and computational semantics (* SEM), volume 2: proceedings of the seventh international workshop on semantic evaluation (SemEval 2013)*, pages 10–14.

Asia J Biega, Peter Potash, Hal Daumé, Fernando Diaz, and Michèle Finck. 2020. Operationalizing the legal principle of data minimization for personalization. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 399–408.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Carlos Busso, Murtaza Bulut, Shrikanth Narayanan, J Gratch, and S Marsella. 2013. Toward effective automatic recognition systems of emotion in speech. *Social emotions in nature and artifact: emotions in human and human-computer interaction*, 7(17):110–127.

A Conneau. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Roddy Cowie, Ellen Douglas-Cowie, Nicolas Tsapatsoulis, George Votsis, Stefanos Kollias, Winfried Fellenz, and John G Taylor. 2001. Emotion recognition in human-computer interaction. *IEEE Signal processing magazine*, 18(1):32–80.

Giovanni Da San Martino, Alberto Barron-Cedeno, and Preslav Nakov. 2019. Findings of the nlp4if-2019 shared task on fine-grained propaganda detection. In *Proceedings of the second workshop on natural language processing for internet freedom: censorship, disinformation, and propaganda*, pages 162–170.

Cristian Danescu-Niculescu-Mizil, Moritz Sudhof, Dan Jurafsky, Jure Leskovec, and Christopher Potts. 2013. A computational approach to politeness with application to social factors. *arXiv preprint arXiv:1306.6078*.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*, volume 11, pages 512–515.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. 2015. Efficient and robust automated machine learning. *Advances in neural information processing systems*, 28.

Eileen Fitzpatrick, Joan Bachenko, and Tommaso Fornaciari. 2015. *Automatic detection of verbal deception*. Morgan & Claypool Publishers.

Diman Ghazi, Diana Inkpen, and Stan Szpakowicz. 2015. Detecting emotion stimuli in emotion-bearing sentences. In *Computational Linguistics and Intelligent Text Processing: 16th International Conference, CICLing 2015, Cairo, Egypt, April 14-20, 2015, Proceedings, Part II 16*, pages 152–165. Springer.

Thiago S Guzella and Walmir M Caminhas. 2009. A review of machine learning approaches to spam filtering. *Expert Systems with Applications*, 36(7):10206–10222.

Ivan Habernal and Iryna Gurevych. 2016. What makes a convincing argument? empirical analysis and detecting attributes of convincingness in web argumentation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1214–1223.

Xin He, Kaiyong Zhao, and Xiaowen Chu. 2021. Automl: A survey of the state-of-the-art. *Knowledge-based systems*, 212:106622.

Tommi Jauhiainen, Marco Lui, Marcos Zampieri, Timothy Baldwin, and Krister Lindén. 2019. Automatic language identification in texts: A survey. *Journal of Artificial Intelligence Research*, 65:675–782.

Aditya Joshi, Pushpak Bhattacharyya, and Mark J Carman. 2017. Automatic sarcasm detection: A survey. *ACM Computing Surveys (CSUR)*, 50(5):1–22.

Douwe Kiela, Suvrat Bhooshan, Hamed Firooz, Ethan Perez, and Davide Testuggine. 2019. Supervised multimodal bitransformers for classifying images and text. *arXiv preprint arXiv:1909.02950*.

Mikhail V Koroteev. 2021. Bert: a review of applications in natural language processing and understanding. *arXiv preprint arXiv:2103.11943*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. *arXiv preprint arXiv:1707.07045*.

Han Liu, Xiaotong Zhang, Lu Fan, Xuandi Fu, Qimai Li, Xiao-Ming Wu, and Albert YS Lam. 2019. Reconstructing capsule networks for zero-shot intent classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4799–4809.

William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. Model cards for model reporting. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 220–229.

Bo Pang, Lillian Lee, et al. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in information retrieval*, 2(1–2):1–135.

MD Pietro. 2020. Text classification with nlp: Tf-idf vs word2vec vs bert. *Preprocessing, Model Design, Evaluation, Explainability for Bag-of-Words, Word Embedding, Language models, Last accessed*, 4(02):2021.

Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *International workshop on semantic evaluation*, pages 19–30.

N Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Anthony Rios and Ramakanth Kavuluru. 2018. Few-shot and zero-shot multi-label learning for structured label spaces. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2018, page 3132. NIH Public Access.

Parisa Safikhani and David Broneske. 2023a. Enhancing autonlp with fine-tuned bert models: An evaluation of text representation methods for autopytorch. *Available at SSRN 4585459*.

Parisa Safikhani and David Broneske. 2023b. Enhancing autonlp with fine-tuned bert models: An evaluation of text representation methods for autopytorch. *Computer Science & Information Technology (CS & IT)*, 13:23–38.

Matthew Shardlow. 2013. A comparison of techniques to automatically identify complex words. In *51st annual meeting of the association for computational linguistics proceedings of the student research workshop*, pages 103–109.

Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter*, 19(1):22–36.

Amit Singhal et al. 2001. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24(4):35–43.

Efstathios Stamatatos, Nikos Fakotakis, and George Kokkinakis. 2000. Text genre detection using common word frequencies. In *COLING 2000 Volume 2: The 18th International Conference on Computational Linguistics*.

James Thorne and Andreas Vlachos. 2018. Automated fact checking: Task formulations, methods and future directions. *arXiv preprint arXiv:1806.07687*.

Alexander Tornede, Difan Deng, Theresa Eimer, Joseph Giovanelli, Aditya Mohan, Tim Ruhkopf, Sarah Segel, Daphne Theodorakopoulos, Tanja Tornede, Henning Wachsmuth, et al. 2023. Automl in the age of large language models: Current challenges, future opportunities and risks. *arXiv preprint arXiv:2306.08107*.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2018. Semeval-2018 task 3: Irony detection in english tweets. In *Proceedings of the 12th international workshop on semantic evaluation*, pages 39–50.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Lucas Zimmer, Marius Lindauer, and Frank Hutter. 2021. Auto-pytorch: Multi-fidelity metalearning for efficient and robust autodl. *IEEE transactions on pattern analysis and machine intelligence*, 43(9):3079–3090.