Bridging the Safety-Specific Language Model Gap: Domain-Adaptive Pretraining of Transformer-Based Models Across Several Industrial Sectors for Occupational Safety Applications

Abid Ali Khan Danish [a], Snehamoy Chatterjee [b]

[a] Ph.D. Candidate, Department of Geological and Mining Engineering and Sciences, Michigan Technological University, MI, USA.

[b] Assistant Professor, Department of Geological and Mining Engineering and Sciences, Michigan Technological University, MI, USA.

Abstract

Occupational safety remains a persistent global challenge despite advancements in regulatory frameworks and safety technologies. Unstructured incident narratives, such as accident reports and safety logs, offer valuable context for understanding workplace hazards but are underutilized due to the gap in the safety-specific language models. This study addresses that gap by adapting pretrained transformer-based models (BERT and ALBERT) to the occupational safety domain through Domain-Adaptive Pretraining (DAPT). We construct a large-scale, multi-source corpus comprising over 2.4 million documents spanning several industrial sectors, including mining, construction, transportation, and chemical processing, augmented with safety-related academic abstracts to preserve general linguistic understanding and mitigate catastrophic forgetting. Using this corpus, we develop two domain-adapted models, safetyBERT and safetyALBERT, through continual pretraining on the masked language modeling objective. Intrinsic evaluation using pseudo-perplexity (PPPL) demonstrates substantial improvements, with safetyBERT and safetyALBERT achieving 76.9% and 90.3% reductions in PPPL, respectively, over their general-domain counterparts. Extrinsic evaluation on the Mine Safety and Health Administration (MSHA) injury dataset across three classification tasks (accident type, mining equipment, and degree of injury) demonstrated consistent performance improvements, with both models outperforming baseline BERT and ALBERT as well as the larger Llama 3.1-8B model, with safetyALBERT achieving competitive results despite its parameter-efficient design. To further assess generalization in low-resource settings, these models were evaluated on the small-scale Alaska insurance claim dataset from mining industry across two classification tasks - claim type and injured body part. Both safetyBERT and safetyALBERT maintained strong performance under this constraint, demonstrating the value of domain adaptation for data-constrained environments. Additionally, multi-task classification on the MSHA dataset using safety domain models showed improved generalization and more balanced performance across underrepresented classes. These findings confirm that DAPT effectively enhances language understanding in safety-critical

domains while enabling scalable, resource-efficient deployment. This work lays the foundation for integrating domain-adapted natural language processing (NLP) systems into occupational health and safety management frameworks.

Keywords: Domain Adaptive Pretraining, Occupational Safety Language Model, Bidirectional Encoder Representations from Transformers, A Lite Bidirectional Encoder Representations from Transformers, Pseudo Perplexity, Accident Classification, Multi-Task Learning

1. Introduction

In recent years, occupational health and safety has advanced significantly, yet workplace incidents remain a global concern. The International Labor Organization reports approximately 2.3 million annual work-related fatalities and 337 million non-fatal injuries worldwide, imposing economic burdens of roughly 4% of global GDP [1]. Despite improvements across various high-risk sectors including mining, construction, transportation, and manufacturing, workplace incidents continue to pose significant challenges across industries [2][3][4]. This underscores the necessity of improving workplace safety through advanced analytical approaches.

Addressing this persistent workplace safety challenge requires the identification of accident trends and patterns through detailed analysis of historical incident and injury data. Traditional analytical methods have largely focused on structured datasets related to past incidents [5]. However, the significance of unstructured data, particularly text-based incident reports, is gaining increasing attention. These reports contain rich, contextual information about incident circumstances, contributing factors, and outcomes that structured data alone cannot capture [6]. Substantial efforts have been made to analyze historical incident data through various techniques across sectors to improve workplace health and safety, including resources [6] [7], construction [8] [9] [10], aviation [11] [12], transportation [13] [14], and manufacturing [15]. Despite these efforts, the potential of unstructured incident narratives for occupational safety enhancement remains underexplored.

Natural Language Processing (NLP) offers powerful tools for extracting meaningful information from unstructured textual data. The field has evolved significantly in the last decade - transforming from simple rule-based methods to advanced Machine Learning (ML) and deep learning models [16]. Several studies have explored textual incident data using traditional statistical language modeling approaches such as term frequency-inverse document frequency (TF-IDF) [17] and N-grams [18] for occupational health and safety applications using ML methods. These studies highlight the effectiveness of employing textual data and statistically based NLP models in enhancing hazard management across

various domains, including road safety [17], mining [19], aviation [20], and construction [21]. However, these statistical techniques fall short in capturing the full semantic richness and contextual nuances of language, which are crucial for a deeper understanding of workplace incident data [22].

To improve contextual understanding, more advanced methods were introduced, such as Word2Vec and GloVe for word embedding extraction [23][24]. These embeddings serve as input features for text classification tasks using ML models [6][17][25][26]; and deep learning architectures like Convolution Neural Network (CNN), Deep Neural Network (DNN), Recurrent Neural Network (RNN), and hybrid models [27][28]. Several researchers explored these embeddings extractions techniques for safety analysis in construction [29][30][31], and railroad domain [27]. In the construction sector, *Zhang (2013)* and *Luo et al. (2023)* demonstrated the utility of Word2Vec embeddings combined with ML models and CNN for automatic classification of workplace incidents [29][30]. Similarly, *Gupta et al. (2022)* underscored the significance of GloVe embeddings in occupational incident analysis by employing a context connotative network (CCNet) for incident cause classification [31]. *Heidarysafa et al. (2018)* explored GloVe and Word2Vec for railroad incident analysis, revealing their effectiveness when coupled with deep learning methods for classifying incident causes [27]. While these studies highlight the capability of static embeddings in capturing semantic relationships, they still could not address the full complexity of language, where deep understanding is critical for comprehensive occupational health and safety applications [22].

The evolution of deep learning-based language models, such as RNNs, Long Short-Term Memory (LSTM) networks [32], and Gated Recurrent Units (GRUs) [33], marked significant progress. However, these architectures struggled with challenges like gradient vanishing and exploding, limiting their ability to process long sequences effectively [34][35]. A pivotal advancement came with the development of the attention mechanism [36], revolutionized by the seminal paper "Attention Is All You Need" [37]. This innovation led to Transformer architecture, fundamentally changing contextual language modeling through powerful pretrained language models (PLMs) like Generative Pretrained Transformers (GPT) [38] and Bidirectional Encoder Representations from Transformers (BERT) [39]. These PLMs gain a broad understanding of language through extensive pre-training and acquiring a foundational grasp of grammar, syntax, and common vocabulary, addressing the limitations of non-contextual language models [40][41]. GPT, a generative model that predicts the probability of word sequences, excels in text-generation tasks [42][38]. Meanwhile, BERT specializes in deep bidirectional text understanding through Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) [39]. While GPT models, particularly GPT-3 and above, perform in-context learning using single or multi-shot approaches with appropriate

prompts, their effectiveness heavily relies on prompt engineering. Despite the success of in-context learning in specific applications, fine-tuning remains the more practical approach for task-specific adaptation of language models, as it directly adjusts model parameters to optimize performance for specific downstream tasks [39].

Through this finetuning approach, BERT and similar PLMs have consistently outperformed non-contextual models across various NLP tasks, including text classification [43], named entity recognition [44], document classification [45], information retrieval [46], text summarization [47], and question-answering [48]. The fine-tuning approach has proven particularly effective in workplace health and safety applications, where researchers have adapted pre-trained BERT models to domain-specific downstream tasks. These applications include incident severity classification [22][49][50][51], extraction of causes and consequences [52][53], prediction of injury leave [54], and entity recognition [55][56], demonstrating BERT's versatility in understanding occupational safety terminology and contexts.

In the construction industry, *Hassan et al. (2022)* presented an innovative approach to classify construction injury narratives by framing it as a question-answering task using pretrained BERT. This method, using the Occupational Safety and Health Administration (OSHA) data, yielded state-of-the-art results in classifying work activity, incident type, injury type, and injury severity - indicating its potential for real-time hazard management [49]. *Goldberg (2021)* utilized word embeddings for categorizing workplace incidents based on injury type, source, and severity, demonstrating the effectiveness of pretrained BERT embeddings combined with ML techniques for automatic classification across different domains, including mining [22]. *Yuan and Wang (2021)* showed the effectiveness of integrating pretrained BERT with Recurrent-CNN for classifying imbalanced traffic incident texts [50], while *Oliaee et al. (2023)* used pretrained BERT to classify crash severity from traffic reports with high predictive accuracy (83.65%) [51]. For extracting causes and consequences, *Linardosa et al. (2022)* fine-tuned pretrained BERT for predicting industrial and railroad incident consequences using multilabel classification, achieving a weighted F1-score of 0.80 [52]. *Song et al. (2022)* combined pretrained BERT with DNN for railroad incident cause classification, achieving an average weighted F1-score of 0.89 [53]. *Ramos et al. (2022)* used pretrained BERT with a Multilayer Perceptron classifier to predict injury leave likelihood with 73.5% accuracy [54]. *Zu et al. (2022)* employed named entity recognition combining Soft Lexicon with a BERT-Transformer-CRF framework to extract risk factors from chemical incident reports [56]. These studies collectively demonstrate the effectiveness of pre-trained BERT models across different industrial sectors for occupational health and safety applications.

Despite these successes, contextual PLMs trained on general texts lack the specialized vocabulary and concepts needed for meaningful predictions in specific domains. This gap has led to retraining language models from scratch with domain-specific texts, resulting in specialized models like BioBERT [57], SciBERT [58], GatorTron [59], CancerBERT [60], KTL-BERT [61], and FinBERT [62][63], which offer improved semantic understanding in biomedical, scientific, healthcare, cancer, tax, and financial fields, respectively. While these domain-specific models demonstrate remarkable performance, retraining transformer-based models like BERT demands substantial computational resources and data, restricting broader application.

To address these challenges, researchers have adopted continual training of PLMs on domain-specific data in construction [64],cybersecurity [65], abusive speech [66], law [67], architecture-engineering-construction [68], and geoscience [69]. In occupational health and safety applications, studies demonstrate this technique's effectiveness in aviation [70][71] and mining domain [72]. These efforts highlight how continual training in domain-adaptive training enables pretrained BERT to acquire domain-specific linguistic nuances while providing equivalent performance compared to pretraining from scratch [73][74][75][76].

Even with continual training being less computationally intensive than training from scratch, it still requires substantial resources. To enhance efficiency, the A Lite BERT (ALBERT) model was introduced, utilizing the same architecture as BERT but with parameter sharing across layers, significantly reducing memory consumption and accelerating training. According to *Lan et al. (2020)*, compared to equivalent BERT models, ALBERT achieves higher data throughput due to decreased communication and computation overhead, with ALBERT-large being approximately 1.7 times faster in processing data than BERT-large. Despite having only about 70% of BERT's parameters, ALBERT-XXL outperforms BERT-large on several representative downstream tasks, including question answering and text classification [77]. This approach has been explored in the different domains, demonstrating effectiveness in enhancing domain-specific language understanding compared to pretrained BERT models [78][79][80].

However, current applications of continual domain-adaptive training for BERT and ALBERT in occupational safety have been limited to specific industries like aviation [70], and mining [72], restricting their broader applicability across sectors. Both pretrained BERT and ALBERT models present promising opportunities for domain adaptation in occupational safety contexts, with ALBERT offering additional efficiency advantages. These characteristics make them ideal candidates for developing language models deployable across diverse industrial sectors, i.e., mining, construction, railroad, and road accidents with varying computational constraints, potentially enabling more widespread implementation of

advanced NLP techniques in workplace safety applications. *Table 1* provides the details of developed domain-adaptive BERT-based models using different training techniques discussed above.

Table 1: Domain-adopted models using several approaches from literature

Model	**Base Model**	**Domain Corpus**	**Continual Training**	**Training from Scratch**
SciBERT [58]	BERT	Biomedical & Computer science	×	✓
BioBERT [57]	BERT	Biomedical	×	✓
BioELECTRA [81]	ELECTRA	Biomedical	×	✓
BioM-ALBERT & BioM-ELECTRA & BioM-BERT [82]	ALBERT, ELECTRA & BERT	Biomedical	×	✓
CancerBERT [60]	BERT	Cancer	✓	×
KTL-BERT [61]	DistilRoBERTa	Korean law	✓	×
MentalBERT & MentalRoBERTa [83]	BERT & RoBERTa	Mental healthcare	✓	×
FinBERT [63]	BERT	Financial	✓	×
HateBERT [66]	BERT	Abusive language	✓	×
CySecBERT [65]	BERT	Cybersecurity	✓	×
BioALBERT [84]	ALBERT	Clinical & Biomedical	✓	×
Aviation-BERT [70]	BERT	Aviation	✓	×
MineBERT [72]	BERT	Mining accidents	✓	×
DistilBERT for Geoscience [69]	DistilBERT	Geoscience	✓	×
BERT for Medical [85]	BERT	Medical	✓	✓
LEGAL-BERT [67]	BERT	Law	✓	✓
BioMegaTron [75]	BERT	Biomedical	✓	✓
BERT for Finance [76]	BERT & ELECTRA	Finance	✓	✓
BERT for biomedical [86]	BERT	Biomedical	✓	✓

Moreover, in the context of workplace health and safety, accurate classification of accident type, equipment involved, and degree of injury is crucial for comprehensive risk assessment and targeted prevention strategies. Several studies demonstrate the significant impact of classifying accidents based on accident type, equipment type, and injury severity in safety management. *Kazan & Usmen (2018)* analyzed earthmoving equipment accidents, finding that inadequate safety training and missing protective systems increased fatality odds

[87]. *Chi et al. (2012)* examined 9,358 construction accidents, identifying relationships between risk factors, accident types, and injury severity to develop strategic prevention plans [88]. *Garcia Cuenca et al. (2018)* compared machine learning techniques for classifying traffic accidents and predicting injury severity [89]. *Shrestha et al. (2020)* proposed using accident investigation reports as leading indicators of safety, employing text classification to extract crucial information about injury precursors, energy sources, accident types, and severity [90]. These studies highlight the importance of understanding accident characteristics and risk factors to prioritize safety measures and develop effective accident prevention strategies. Their success underscores the value of accurate classification models in the occupational safety domain. However, achieving optimal classification performance demands language models with comprehensive understanding of the safety domain context, and domain-adapted models specifically trained on occupational safety data address this gap more effectively than general-purpose alternatives. Recent studies validate the effectiveness of domain-adapted language models in the occupational safety domain: *Song et al. (2024)* demonstrated a KoBERT-based model that achieved 93.1% accuracy in classifying accident occurrence types, while *Ansari et al. (2024)* provided empirical evidence that domain-specific fine-tuned models outperform general-purpose models in critical safety applications including compliance assessment and incident reporting analysis [91][92]. While these studies have advanced classification capabilities, they frequently classify safety variables independently (e.g., injury severity, accident type), overlooking critical interrelationships between these factors [51][54]. This limitation creates an opportunity for multitask learning approaches that can capture interconnected effects [22]. This comprehensive approach captures both individual factors and their combined dynamics, providing more nuanced insights for targeted prevention strategies.

Building on these insights, this study focuses on domain adaptation of pretrained BERT and ALBERT models for workplace health and safety language understanding. We developed safetyBERT and safetyALBERT with the aim of creating models applicable across various industrial sectors. To evaluate their effectiveness, we implemented both single-task classification and multitask learning frameworks. Both approaches were fine-tuned on downstream classification tasks using mining safety data: accident type, mining equipment, and degree of injury classification using MSHA injuries data, as well as claim type and injured body part classification using Alaska's insurance claim data from mining industry. Performance comparisons with the larger llama3 8B model established benchmarks for occupational safety domain language understanding, demonstrating the practical utility of our approach in addressing the computational challenges of domain-specific language modeling while maintaining high performance on safety-critical classification tasks. The comparison with llama3 8B is particularly significant given that larger models like llama3

typically contain billions of parameters requiring substantial computational resources. As *Touvron et al. (2023)* demonstrate, while large language models with 8B+ parameters can achieve superior performance on general domain tasks, they require exponentially greater computational resources during both training and inference compared to lightweight models like BERT and ALBERT, with training costs often exceeding 30-40 times higher for comparable performance in specialized domains [93][94].

2. Research Methodology

This section outlines the research methodology employed to explore the domain-specific adaptation of transformer-based language models for occupational safety applications. The central premise of this study is that PLMs, initially trained on a general corpus will benefit from DAPT on an occupational safety-specific domain corpus. The DAPT phase, depicted in *Figure 1*, is hypothesized to enhance the models' comprehension of domain-specific nuances, thereby optimizing their performance in downstream NLP tasks. DAPT of language models to a specific domain is a well-established method to achieve advanced domain-specific language modeling capabilities [74]. This prospects lead us to expect that the occupational safety domain will significantly benefit from a DAPT of PLMs for various domain specific tasks [68][73][86].

The proposed workflow consists of three parts: (1) Domain corpora development - This phase involves the development of a comprehensive multi-source occupational safety corpus, subject to specific data cleaning and pre-processing techniques. Further details of this stage are provided in *Section 2.1*. (2) Pre-train PLMs on domain corpora - This critical second phase transforms general-purpose language models into the occupational safety domain model through continual training on our specialized corpus. We implement MLM domain-adaptive pre-training approach that enhances model understanding while preserving general language capabilities. A detailed exposition of this phase is available in *Section 2.2*. (3) The fine-tuning of pre-trained domain models - This final phase evaluates the practical utility of our domain-adapted models by applying to specific occupational safety tasks. We measure their performance improvements in domain-specific applications. This phase encompasses the development and assessment of models for classification tasks using the finetuning approach. A more extensive explanation of this step is provided in *Section 2.3*.

The entirety of the workflow is implemented in Python 3.11 and leverages a multitude of Python packages, including Transformers, PyTorch, pandas, scikit-learn, Matplotlib, and NumPy to facilitate the development of our models. All experiments were conducted on Ubuntu 22.04.5 LTS 64-bit with an NVIDIA A100-SXM4-40GB GPU, AMD Epyc-Milan processor × 32, and CUDA Version 12.1.

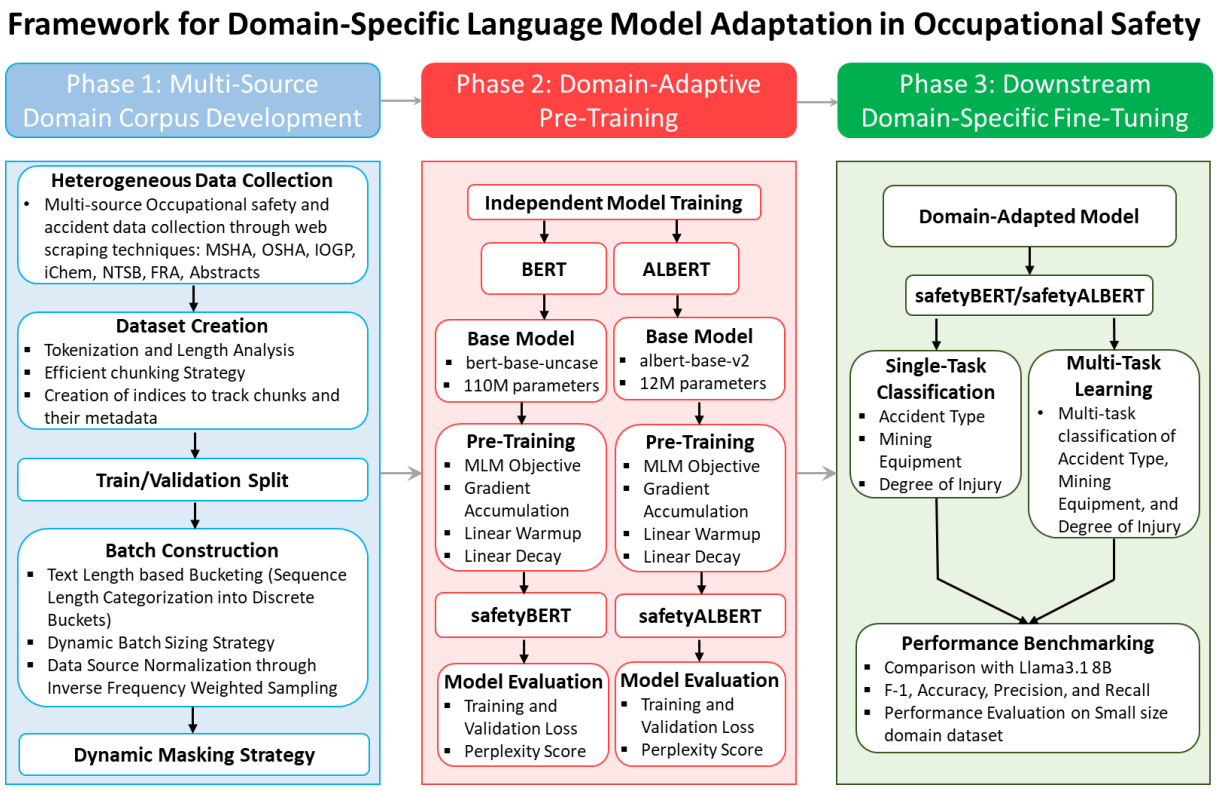


Figure 1: Methodological Framework for Domain-Specific Language Model Adaptation in Occupational Safety, illustrating the three-phase approach: multi-source corpus development (left), domain-adaptive pre-training of BERT and ALBERT models (center), and downstream task evaluation through single-task and multi-task classification with performance benchmarking against Llama 3.1-8B (right).

2.1. Multi-Source Domain Corpus Development

2.1.1. Corpus Development for DAPT of PLMs

This research establishes a multi-source occupational safety corpus for DAPT of PLMs. Developing a comprehensive domain-adaptive language model for occupational safety requires diverse textual data that captures the full spectrum of linguistic patterns, terminology, and contexts across industrial sectors. The primary challenge lies in collecting, standardizing, and integrating data from disparate sources with varying formats, structures, and accessibility constraints. This section addresses how we systematically gather heterogeneous occupational safety data to build a corpus that represents both academic and practical aspects of occupational safety.

We implemented a multi-source extraction strategy targeting seven distinct data sources spanning academic literature, regulatory agencies, and industry organizations. For each source, we developed a specialized extraction methodology based on data accessibility, format, and structure, as outlined in *Table 2*. This diversified approach ensured comprehensive domain coverage while maintaining data quality. Two primary extraction

frameworks were developed: 1) Academic Abstract Collection Framework, a systematic API-based extraction system for retrieving safety-related academic literature from major publishers, and 2) Accident Narrative Collection Framework, a multi-method approach for extracting incident narratives/reports from regulatory and industry sources.

Table 2: Extraction Strategies by Data Source, detailing the multi-source data collection methodology implemented across seven distinct data sources spanning academic and regulatory domains, with corresponding extraction techniques tailored to each source's format and accessibility.

Data Source	Domain	Data Retrieval Source	Data Format	Extraction Technique	Implementation Environment
Academic Journals	Occupational Safety	Elsevier, Springer, and Taylor & Francis	API/Web	API Client & Web Scraping	Python 3.11, Requests, Selenium
FRA	Railroad	https://railroads.dot.gov/safety-data/new-safety-data-site	Structured Data	Direct Download	Web Browser, Excel
IChemE	Chemical Industry	https://www.icheme.org/knowledge-networks/knowledge-resources/safety-centre/resources/accident-data/	PDF Reports	PDF Text Extraction	Python 3.11, PDFMiner
IOGP	Oil and Gas Industry	https://www.iogp.org/bookstore/product-category/data-series/	PDF Reports	PDF Text Extraction	Python 3.11, PDFPlumber
NTSB	Transportation	https://www.ntsb.gov/safety/data/Pages/Data_Stats.aspx	Structured Data	Direct Download	Web Browser, Excel
OSHA	Construction Industry	https://www.osha.gov/ords/imis/accidentsearch.html	Web Database	Web Scraping	Python 3.11, Selenium
MSHA	Mining Industry	https://www.msha.gov/data-and-reports/mine-data-retrieval-system	Web Database	Direct Download	Web Browser, Excel

Academic journal abstracts were collected from major scholarly publishers, including Elsevier, Springer, and Taylor & Francis. These academic journal abstracts were incorporated into our corpus to expose the model to terminologies associated with the broader occupational safety context. Previous research in domain-specific language modeling has demonstrated that models trained exclusively on specialized texts may lead to catastrophic forgetting, exhibiting reduced performance on general language tasks [95] [96] [97]. The integration of academic abstracts alongside incident narratives and reports maintains a crucial balance between domain-specific terminology and general linguistic constructions within the occupational safety field. Similarly, detailed incident narratives were collected from multiple regulatory and industry sources: Federal Railroad Administration (FRA), MSHA, OSHA, Institution of Chemical Engineers (IChemE), International Association of Oil & Gas Producers (IOGP), and National Transportation Safety Board (NTSB). These narratives provided rich, domain-specific incident descriptions across diverse industries.

2.1.1.1. Data Collection

For academic journals' abstracts, we implemented a consistent API-based extraction approach, illustrated in *Figure 2*. The Elsevier corpus was developed using their Text and Data Mining API with 133 safety-related keywords. Search terms were transformed into API-compatible query strings with necessary pagination parameters to ensure comprehensive data retrieval across all available results. Similar API-based approaches were employed for Springer, and Taylor & Francis journals, utilizing their respective APIs with appropriate authentication and query parameters. The extraction followed a two-stage pattern: first

retrieving article metadata (including DOIs) by parsing JSON/XML responses, then using these DOIs to retrieve specific abstract content through targeted API calls. All extracted text underwent normalization processes, including character encoding standardization, whitespace regularization, and artifact removal to ensure corpus consistency.

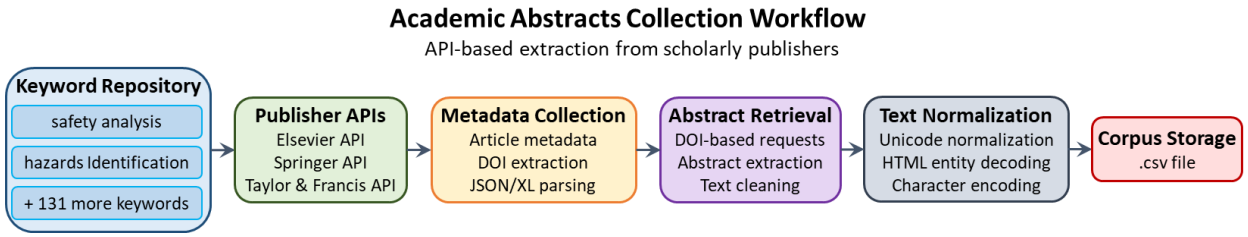


Figure 2: Academic Abstracts Collection Workflow, illustrating systematic extraction of safety-related abstracts from scholarly publishers through API-based workflows, including keyword repositories, publisher APIs, metadata collection, abstract retrieval, and text normalization stage.

Similarly, the accident narrative corpus was developed through several methodologies tailored to each source format, as illustrated in *Figure 3*. The FRA's safety portal was accessed to obtain structured accident narratives including Rail Equipment Accident/Incident Data (Form 54), Injury/Illness Summary data (Forms 55 and 55A), Highway-Rail Grade Crossing Accident Data (Form 57), and Crossing Inventory Data (Form 71). The NTSB aviation accident database was acquired in Microsoft Access format (.mdb) and processed using pyodbc and mdbtools libraries to extract accident reports.

The OSHA accident corpus required a more sophisticated approach using a comprehensive web scraper that systematically accessed OSHA's accident search system through an alphabetical keyword approach. This multi-level extraction first identified all keywords starting with each alphabet from A to Z, then retrieved all associated accident entries. The extractor incorporated dynamic waiting mechanisms to address server response variability and employed XPath-based content targeting to isolate narrative sections. MSHA accident narratives were obtained directly from their data retrieval database in text format.

Similarly, to obtain accident narratives from the IChemE Safety Centre repository and IOGP, accident reports were collected in PDF format and subsequently processed to extract accident narrative sections. This involved specialized extraction techniques using the PDFMiner and PDFPlumber libraries, with a focus on identifying narrative sections through positional word analysis to reconstruct coherent text from complex PDF layouts. All extracted data underwent consistent preprocessing including Unicode normalization, special character handling, and whitespace standardization across all sources to ensure corpus consistency.

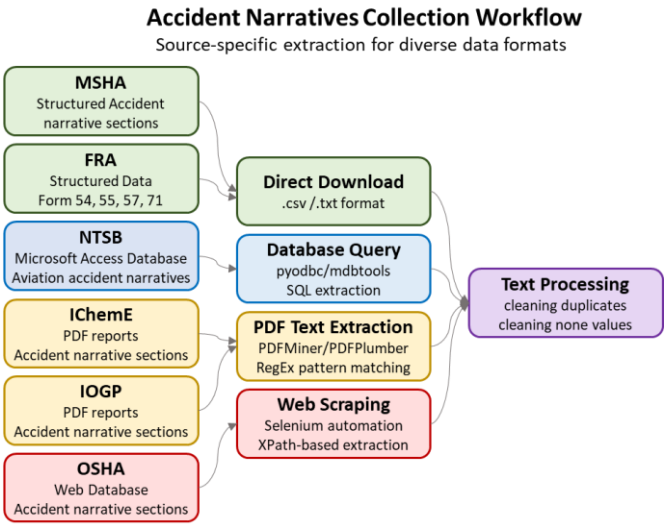


Figure 3: Accident Narratives Collection Workflow, illustrating source-specific extraction methodologies for diverse data formats across regulatory and industry sources, including structured downloads, database queries, PDF text extraction, and web scraping techniques for comprehensive accident narrative acquisition.

The corpus development process was implemented using Python 3.11, leveraging specialized libraries for different extraction tasks. Data manipulation employed Pandas and NumPy for structured data handling. Web interaction utilized Requests for API communication, Selenium for dynamic web content interaction, and the BeautifulSoup for HTML parsing. PDF processing leveraged PDFMiner and PDFPlumber for content extraction based on document structure complexity. Database interaction for the NTSB Access database utilized pyodbc with appropriate ODBC drivers for structured query execution. Text processing incorporated NLTK and Regular Expressions for pattern matching and cleaning.

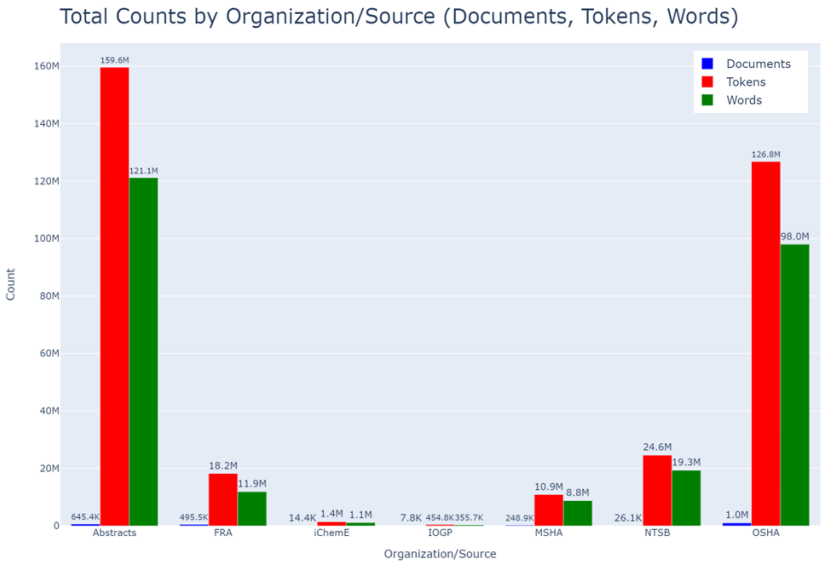


Figure 4: Distribution of Documents, Tokens, and Words by source, visualizing the contribution of each data source to the occupational safety corpus, with academic abstracts and OSHA narratives representing the largest token volumes despite significant differences in document counts.

The resulting corpus exhibits significant diversity in both size and content type across sources, as illustrated in *Figure 4*. *Table 3* provides all the statistical details of the collected corpus, showing that academic journal abstracts constitute 26.2% of documents but 46.7% of total tokens, reflecting their linguistic density. Conversely, accident narratives comprise 73.8% of documents but only 53.3% of total tokens in the corpus as illustrated in *Figure 5*. This distribution creates a relatively balanced text collection combining general safety-related academic terminology with specific accident descriptions and varied linguistic patterns. This balanced composition ensures the corpus captures both the technical terminology and practical language patterns essential for effective domain adaptation of language models in the occupational safety context.

Table 3: Multi-Source Occupational Safety Corpus Distribution, showing document counts, token and word statistics across academic and regulatory sources.

Data Source	Content Type	Documents	Documents %	Tokens	Tokens %	Words	Words %	Avg Words/ Doc	Avg Tokens/ Doc	Token/ Word Ratio
Elsevier, Springer, and Taylor & Francis	Research Abstracts	645,402	26.20%	159,568,152	46.7%	121,145,576	46.5%	187.7	247.2	1.32
FRA	Accident Narratives	495,520	20.10%	18,222,963	5.3%	11,898,308	4.6%	28.5	43.7	1.53
ICHEM	Accident Narratives	14,374	0.60%	1,433,145	0.4%	1,137,231	0.4%	79.1	99.7	1.26
IOGP	Accident Narratives	7,840	0.30%	454,758	0.1%	355,749	0.1%	45.4	58	1.28
MSHA	Accident Narratives	248,866	10.10%	10,896,576	3.2%	8,783,547	3.4%	35.3	43.8	1.24
NTSB	Accident Narratives	26,057	1.10%	24,590,329	7.2%	19,344,213	7.4%	742.4	943.7	1.27
OSHA	Accident Narratives	1,021,803	41.50%	126,777,133	37.1%	97,993,041	37.6%	95.9	124.1	1.29

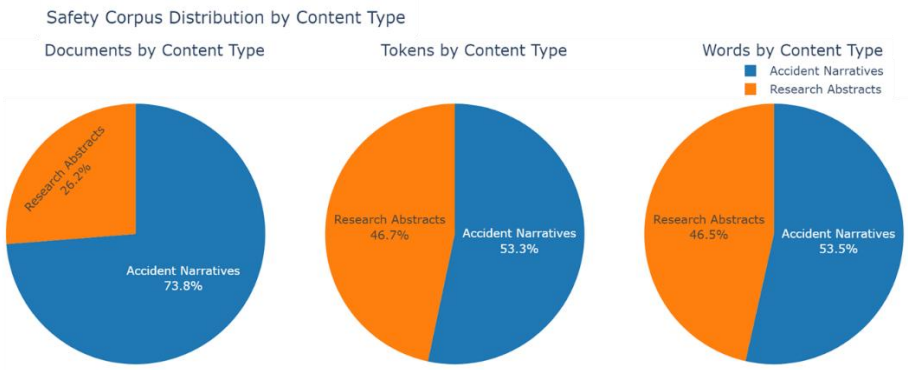


Figure 5: Safety Corpus Distribution by Content Type, comparing the proportional representation of accident narratives and research abstracts across three metrics: documents (73.8% vs. 26.2%), tokens (53.3% vs. 46.7%), and words (53.5% vs. 46.5%), illustrating the balanced linguistic composition despite document count disparities.

2.1.2 Tokenization Approach

Following data collection from heterogeneous sources, our preprocessing pipeline prepared the multi-source occupational safety corpus for domain-adaptive pre-training. The corpus underwent systematic tokenization and input sequence processing to ensure optimal MLM training efficiency. It is a fundamental preprocessing step in NLP that segments text into smaller units (tokens) that serve as input to language models [98]. Effective tokenization balances vocabulary size constraints with the need to represent semantic meaning, particularly for domain-specific terminology found in occupational safety narratives.

For our domain-adaptive pre-training approach, we utilized the existing pre-trained tokenizers corresponding to our base language models (BERT and ALBERT). For BERT-based models, we employed the pre-trained WordPiece tokenizer with a vocabulary size of 30,522 tokens [99]. WordPiece follows a data-driven approach that iteratively merges frequent character sequences to form subword units, efficiently handling out-of-vocabulary terms while maintaining semantic relationships [98]. For ALBERT-based models, we utilized the pre-trained SentencePiece tokenizer with a vocabulary size of 30,000 tokens [77]. The SentencePiece treats input as a raw character sequence and applies byte-pair encoding to learn subword units directly from the corpus, offering language-agnostic tokenization without requiring pre-tokenization [100]. Special tokens ([CLS], [SEP], [MASK], [PAD]) were preserved according to the requirements of the masked language modeling objective.

2.1.3. Sequence Processing and Categorization for Batch Construction

Variable-length inputs present significant challenges for transformer-based models during training. Recent studies have demonstrated that transformer-based models struggle with processing variable sequence lengths due to their self-attention operation, which scales quadratically with sequence length [100]. Similarly, traditional frameworks requiring padding

to maximum length add significant memory and computational overhead for variable-length inputs, with padding-free approaches improving performance significantly [101].

Analysis of the occupational safety corpus revealed substantial sequence length variability across different data sources, as shown in *Figure 6*. Academic abstracts exhibited the highest average token length (247.2 tokens) with maximum lengths reaching 4,345 tokens. Among accident narratives, NTSB reports demonstrated the greatest average length (943.7 tokens) and maximum length (21,548 tokens), while FRA narratives were considerably shorter (43.7 tokens on average). This distribution created a highly heterogeneous corpus with sequence lengths spanning from as few as 3 tokens to over 21,000 tokens.

For processing, an upper limit (512) was established consistent with the standard context window size of BERT and ALBERT models [99][77]. For texts exceeding this limit, an overlapping chunking strategy was implemented with a predefined stride parameter. This approach divides longer documents into overlapping segments (stride) where each segment shares a portion of tokens with adjacent segments. The overlap ensures contextual continuity across chunk boundaries and allows the model to capture cross-segment dependencies. For shorter texts, their complete structure was preserved to maintain contextual integrity.

To address the computational challenges of processing this variable-length corpus, a memory-optimized batching strategy was developed. This approach builds upon efficient sequence handling techniques [102], categorizing sequences into five distinct length buckets: very short (≤64 tokens), short (65-128 tokens), medium (129-256 tokens), long (257-384 tokens), and very long (385-512 tokens). Each category received a dynamically scaled batch size inversely proportional to sequence length, with very short sequences processed in larger batches and very long sequences in smaller batches. Within each batch, sequences were only padded to match the length of the longest sequence in that specific batch, rather than padding to the full 512-token context window, allowing the model to process variable-length inputs efficiently while significantly reducing memory wastage.

The critical implementation in this approach is batch homogeneity, each batch contains sequences exclusively from a single length category. This design minimizes padding waste and optimizes GPU memory utilization while still exposing the model to the full diversity of sequence lengths during training. During training, the algorithm constructs multiple batches for each length category, maintaining batch homogeneity throughout. These batches are then randomly shuffled before training to ensure diversity in sequence exposure while preserving the computational benefits of homogeneous batching.

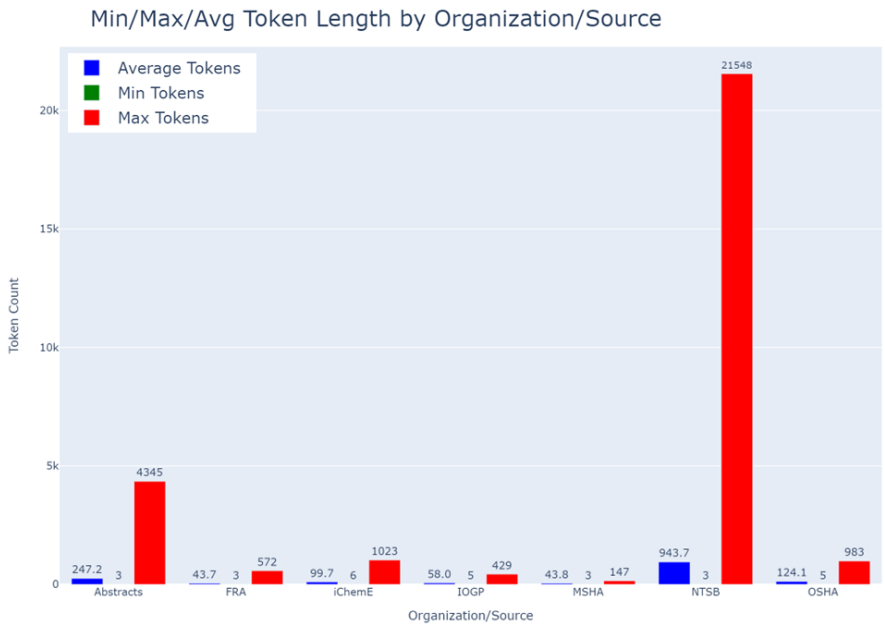


Figure 6: Min/Max/Average Token Length by Organization/Source, highlighting the significant sequence length variability across data sources, with NTSB accident narratives showing the highest maximum length (21,548 tokens) and average length (943.7 tokens), while research abstracts and other industrial narratives demonstrate diverse length distributions requiring optimized processing strategies.

2.1.4. Addressing Source Domain Imbalance

As shown in *Table 3*, the corpus exhibits considerable disparity in source distribution, with OSHA narratives comprising 41.5% of documents, academic abstracts 26.2%, and FRA narratives 20.1%, while sources such as IOGP and IChemE contribute only 0.3% and 0.6% respectively. This imbalance could potentially bias the model toward overrepresented domains and linguistic patterns, diminishing its effectiveness for specialized industry applications.

To mitigate this challenge, our methodology implements a weighted sampling approach, calculating source-specific sampling weights inversely proportional to their frequency in the dataset during batch construction:

$$\omega_s = \frac{N_{total}}{N_s} \tag{1}$$

Where ω_s is the weight for source, N_{total} is the total number of sequences, and N_s is the number of sequences from source s.

These weights can be incorporated into the training process during batch construction, where the probability of including a sequence from source s becomes:

$$P(x_i) \propto \omega_{s_{(i)}} \tag{2}$$

Where $s_{(i)}$ denotes the source of the sequence x_i .

This approach ensures that during the random shuffling of batches before each training epoch, sequences from underrepresented sources are effectively oversampled, giving them proportionally greater representation in the training process. This formulation ensures that sequences from underrepresented sources like IOGP and IChemE contribute proportionally more during training, countering the potential bias from source distribution imbalance and ensuring that specialized industry terminology and patterns are adequately represented in the adapted model. The sequential implementation process of this approach is illustrated in *Figure 7*.

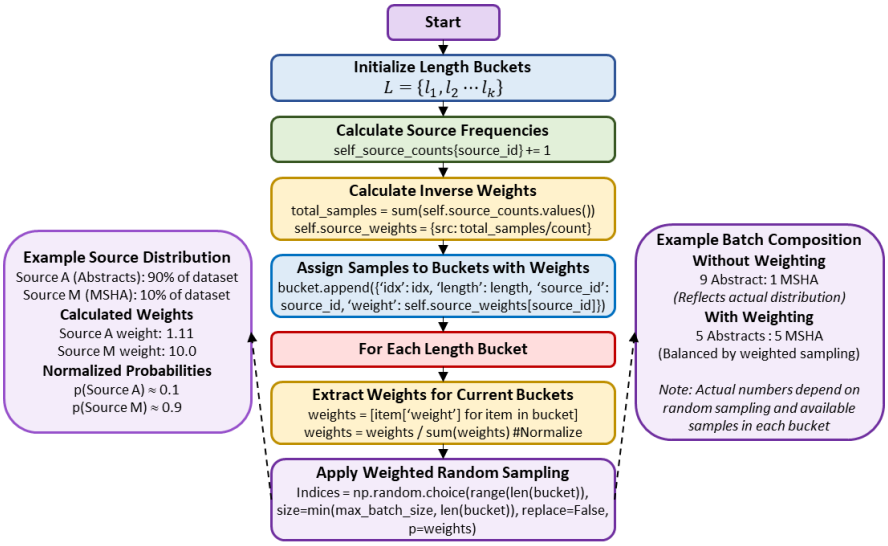


Figure 7: Weighted Sampling Process for Domain Balance, illustrating the algorithm for addressing source domain imbalance through inverse frequency weighting, with example source distribution (left) showing calculated weights and normalized probabilities, and resulting batch composition comparison (right) demonstrating the effect of weighting on source representation.

The complete sequence processing and domain-balanced training process is formalized in *Algorithm 1 as following*, which simultaneously addresses the computational challenges of variable-length sequences and the potential training bias from domain imbalance.

ALGORITHM 1: Domain-Balanced Training via Weighted Sampling
Input: Corpus C = $\{(x_i, s_i)\}_{i=1}^{N}$, where x_i is a token sequence, and s_i is its source
Output: Trained model *M*
1: // Step 1: Compute inverse-frequency sampling weights per source
2: for each unique source *s* in *C* do
3: $N_s \leftarrow$ count of sequences in *C* with source *s*
4: $w_s \leftarrow \lvert C \rvert / N_s$ // Inverse frequency weight
5: end for

6: Normalize weights:
7: for each source s do
8: $p_s \leftarrow w_s / (\sum_{s' \in S} w_{s'})$ // Sampling probability
9: end for
10: Step 2: Group sequences into buckets based on token length
11: for each (x_i, s_i) in C do
12: token_len ← TokenCount(x_i)
13: bucket_idx ← floor(token_len / bucket_size)
14: bucket[bucket_idx] ← bucket[bucket_idx] ∪ {(x_i, s_i)}
15: end for
16: // Step 3: Training with weighted sampling and dynamic batching
17: for epoch = 1 to E do
18: for each bucket b in bucket[] do
19: max_len ← MaxTokenLength(b)
20: batch_size ← DetermineBatchSize(max_len)
21: while b is not empty do
22: β ← ∅ // Initialize empty batch
23: while |β| < batch_size, and b is not empty do
24: Sample (x, s) from b with probability ∝ p_s
25: x_{padded} ← PadToLength(x, max_len)
26: β ← β ∪ {(x_{padded}, s)}
27: b ← b \ {(x, s)}
28: end while
29: UpdateModel(M, β)
30: end while
31: end for
32: end for
33: return M

2.2. DAPT of PLMs on Domain Corpus

This section details our specific methodological approach to DAPT of PLMs for occupational safety. The research presented herein employs pretrained BERT and ALBERT architectures as foundation models for DAPT. The implementation incorporates the MLM for continued pretraining, alongside advanced data handling strategies and optimization techniques that enhance the domain adaptation process. The following subsections provide a comprehensive description of our training methodology and implementation details.

2.2.1. Masked Language Modeling (MLM):

Traditionally, the pretraining of BERT-based models included two main objectives: MLM and Next Sentence Prediction (NSP). However, subsequent research by multiple studies, including the XLNet [103] and ALBERT [77] has critically examined the NSP task and found it to be inconsistent and ineffective. This ineffectiveness of NSP stems from its inherent design limitations, as it conflates two distinct prediction objectives: topic prediction and inter-sentence coherence prediction. The topic prediction component is relatively easy to learn, as it significantly overlaps with the MLM loss. Consequently, the NSP task can yield higher scores even when it fails to effectively learn coherence prediction, essentially introducing noise into the pretraining process [104]. Given these insights, this research adopts the MLM approach for DAPT of the models. The MLM involves masking parts of the input text and training the model to predict these masked tokens. The series of steps constituting the MLM procedure, as diagrammatically represented in *Figure 8*, is elucidated in sequential order.

The first stage involves data preparation, as discussed in *Section 2.1.1***.** The second stage constructs the input embeddings. In BERT architecture, input embeddings are constructed by combining three distinct embedding components: token embeddings, segment embeddings, and position embeddings. As shown in the *Figure 8*, this process differs between architectures - BERT uses full-dimensional embeddings (768-dim) throughout, while ALBERT employs a factorized approach with lower-dimensional embeddings (128-dim) projected to higher dimensions (768-dim), significantly reducing parameter count. This embedding construction encodes safety domain tokens while maintaining positional and structural information critical for understanding accident narratives [77].

These can be represented formally as:

$$E_i^{BERT} = E_{\omega_i}^{token} + E_{s_i}^{seg} + E_{p_i}^{pos} \quad (3)$$

Where:

$E_{\omega_i}^{token} \in \mathbb{R}^d$ is the token embedding for word ω_i in vocabulary *V*

$E_{s_i}^{seg} \in \mathbb{R}^d$ is the segment embedding for segment $s_i \in \{A, B\}$

$E_{p_i}^{pos} \in \mathbb{R}^d$ is the segment embedding for segment $p_i \in \{0,1, \ldots., 511\}$

d = 768 is the hidden dimension size

ALBERT modifies this approach through factorized embedding parameterization:

$$E_i^{ALBERT} = P(E_{\omega_i}^{token-lower} + E_{p_i}^{pos-lower}) \quad (4)$$

Where:

$E_{\omega_i}^{token-lower} \in \mathbb{R}^e$ is the lower-dimensional token embedding

$E_{p_i}^{pos-lower} \in \mathbb{R}^e$ is the lower-dimensional position embedding

$P \in \mathbb{R}^{exd}$ is the projection matrix from embedding dimension to hidden dimension

e = 128 is the embedding dimension (significantly smaller than *d*)

This factorization substantially reduces parameters in the embedding layer while maintaining model representational capacity.

The third stage applies our masking strategy on input sequences, strategically replacing tokens with [MASK] tokens. While the original BERT pretraining used static masking applied once during data preprocessing - potentially leading to overfitting [105], this research implements dynamic masking. Dynamic masking generates a new pattern each time a sequence is processed, which has demonstrated enhanced performance across multiple datasets [105]. Specifically, this implementation selects 15% of input tokens for prediction in each training batch, with 80% replaced by [MASK] tokens, 10% with random vocabulary tokens, and 10% left unchanged. This approach is particularly important for safety terminology where specialized vocabulary requires robust contextual understanding [106].

The fourth stage processes these masked sequences through the transformer encoder architectures, with a notable difference between models. The BERT architecture utilizes independent parameters across all 12 layers (*Figure 8a*), while the ALBERT model employs parameter sharing across layers (*Figure 8b*), dramatically reducing memory requirements while maintaining performance. This architectural difference allows ALBERT implementation to process larger batches with similar computational resources [77].

The final stage implements MLM prediction, where the model predicts the original tokens at masked positions using a Feedforward neural network (FFNN) with softmax activation. As shown at the top of *Figure 8a and Figure 8b*, the models predict probability distributions across their respective vocabularies (30,522-dim for BERT, 30,000-dim for ALBERT) for all tokens. For optimization, we employ cross-entropy loss calculated only on masked positions, as given in the following equation.

$$\mathcal{L}_{MLM} = -\sum_{i \subset M} log\ P(w_i|\tilde{w}) \tag{5}$$

Where w_i is the correct token at masked position *i* and $\tilde{w}$ is sequence with masked tokens. $P(w_i|\tilde{w})$ is the predicted probability of the correct token at position *i*. This loss function is minimized using AdamW optimizer with weight decay and a learning rate for BERT and ALBERT, following recommendations in literature [99] [77].

For both architectures, optimization of the pre-training process involves several advanced techniques. The memory-aware smart batch sampler systematically categorizes documents into length-based buckets with dynamically scaled batch sizes, enabling more efficient GPU memory utilization while ensuring balanced training across document types. Gradient accumulation across multiple forward-backward passes with dynamic batch size

enables training with more efficient GPU utilization while maintaining numerical stability. For the ALBERT implementation, automatic mixed precision training through PyTorch's AMP functionality selectively uses FP16 and FP32 computation dynamically, with automatic loss scaling to prevent gradient underflow - reducing memory requirements while accelerating computation [107]. The implementation ensures balanced representation across all seven data sources through weighted sampling, preventing larger datasets from dominating the training signal and improving cross-sector generalization. After completing the pre-training phase with these optimizations, the checkpoint with the lowest validation loss is selected for both BERT and ALBERT models. These domain-adapted models, safetyBERT and safetyALBERT, then undergo evaluation as described in the subsequent sections.

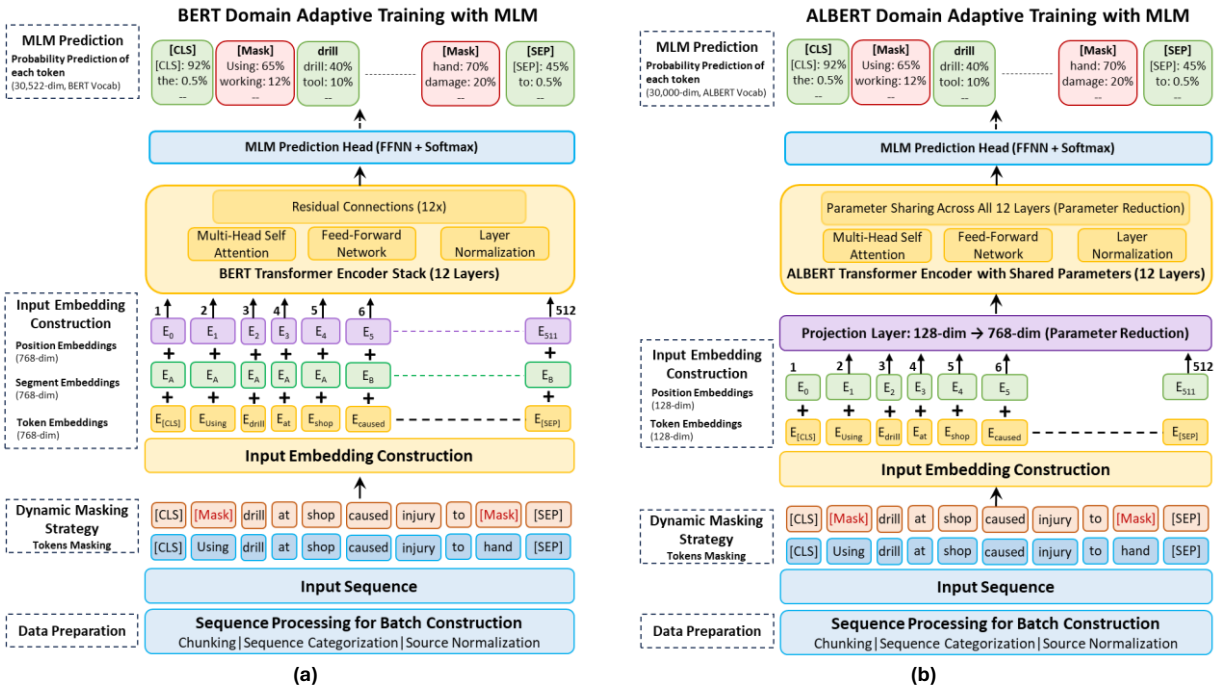


Figure 8: DAPT with MLM objective (a) BERT DAPT, showing the full-dimensional (768-dim) embedding construction and independent parameters across all 12 transformer layers. (b) ALBERT DAPT, illustrating the parameter-efficient approach with factorized embeddings (128-dim) and cross-layer parameter sharing, reducing parameters from 110M to 12M while maintaining the same MLM prediction capabilities.

2.2.2. Domain-Adapted Model Evaluation

This section details the evaluation methodology implemented to measure performance improvements attributable to domain adaptation and to compare domain-adapted models (safetyBERT and safetyALBERT) against their general-purpose counterparts. For this purpose, this research employs both intrinsic and extrinsic evaluation methods. For intrinsic evaluation, pseudo-perplexity (PPPL) is calculated on data that is not seen by models during MLM training, while extrinsic evaluation involves fine-tuning models for downstream classification tasks as discussed in *Section 2.3*.

2.2.2.1. Intrinsic Evaluation

Pseudo-perplexity provides a meaningful metric for assessing bidirectional language models like BERT and ALBERT [108]. Unlike traditional perplexity which relies on left-to-right context in autoregressive models, PPPL uses a masked language modeling approach that better aligns with how bidirectional transformers are trained.

Let $W = (w_1, w_2, \ldots, w_n)$ represent a sequence of tokens in the text corpus. The pseudo-log-likelihood (PLL) for a sequence W with model parameters Θ is formally defined as:

$$PLL(w) = \sum_{t=1}^{|w|} logP(w_t|W \setminus t; \Theta) \tag{6}$$

Where $W\backslash t$ represents all tokens in the sequence except the token at position t, w_t is the token at position t, and Θ represents the model parameters. This calculation requires systematically masking each token individually and measuring the model's probability of predicting the original token given its surrounding bidirectional context.

From this, the *PPPL* is then derived by normalizing over sequence length and applying the exponential function:

$$PPPL(W) = exp\left(-\frac{1}{N} \text{ x } PPL(W)\right) \tag{7}$$

Where N is the number of tokens. To ensure fair comparison between models with different subword tokenization schemes (e.g., between BERT and ALBERT), a word-normalized version ($PPPL_w$) is calculated:

$$PPPL_w(W) = exp\left(-\frac{1}{N_w} \text{ x } PPL(W)\right) \tag{8}$$

Where N_w represents the word count rather than token count.

The implementation of PPPL starts by creating masked copies of each sequence during evaluation by applying the [MASK] token at each position t successively. For each position, it calculates $logP(w_t|W \setminus t; \Theta)$ by replacing w_t with [MASK], passing the masked sequence through the model, and computing the log probability of the original token at the masked position. The token-level calculations are then aggregated to produce sentence-level PLL scores, excluding special tokens ([CLS], [SEP], [PAD], [MASK]) from both masking operations and final PLL calculations to prevent skewing the results. This methodology enables a direct comparison between pretrained models (BERT-base, ALBERT-base) and their domain-adapted variants (safetyBERT, safetyALBERT), with lower PPPL scores serving as an indicator of more effective domain adaptation and reflects the model's improved ability to predict domain-specific terminology and contextual patterns in occupational safety texts.

2.2.2.2. Extrinsic Evaluation

In this study, the extrinsic evaluation framework assesses domain adaptation effectiveness. This methodology involves fine-tuning the domain-adapted models for occupational safety specific classification tasks using a supervised learning approach. Fine-tuning adapts the domain-adapted models (safetyBERT, safetyALBERT) to predict target labels by adding task-specific classification layers while freezing transformer layers. The methodology incorporates a comprehensive evaluation using the MSHA dataset and a relatively small dataset (Alaska mining injuries insurance claim dataset) to examine performance across varying data availability conditions.

The weighted F1 score serves as the primary performance metric, calculated as the harmonic mean of precision and recall with class weights proportional to their frequencies in the dataset:

$$F1_w = \sum_{i=1}^{N} \omega_i \, \mathrm{x} \, F1_i \tag{9}$$

Where ω_i represents the proportion of samples belonging to class *i*, $F1_i$ is the F1 score for class *i*, and *N* is the total number of classes. The weighted F1 score reflects the model's overall performance across all instances, which aligns with the goal to maximize accuracy in practical applications where the distribution of classes mirrors their real-world frequencies. Due to the complexity and diversity of the data, it's common to encounter imbalanced datasets in occupational safety. In such situations, relying solely on accuracy can be misleading since a high accuracy does not always translate to a good model, especially when there's a significant class imbalance. Therefore, the weighted F1 score is used because it effectively balances the trade-off between precision and recall. For multi-task evaluation scenarios, both task-specific metrics and an aggregate performance score quantify the model's capacity for simultaneous handling of interrelated safety classification objectives.

The comparative evaluation structure isolates domain adaptation effects by maintaining identical fine-tuning procedures across pre-trained models (BERT-base, ALBERT-base), domain-adapted variants (safetyBERT, safetyALBERT), and a pretrained large language model (Llama-3.1-8B). Complete implementation details of this evaluation framework are provided in *Section 2.3*.

2.3. Finetuning of Domain-Adapted Models

In this study, the primary objective of finetuning domain models is to evaluate the performance of domain-adapted models on downstream domain-specific classification tasks. This evaluation involves comparing the models' performance before and after DAPT and benchmarking them against available open-source large language model (Llama 3.1-

8B). By employing this methodological approach, this research aims to demonstrate the efficacy of domain-specific model adaptation in improving classification accuracy and contextual understanding. The following sections delineate the research methodology, detailing the corpus development, preprocessing techniques, and the intricate process of fine-tuning domain-adapted models for occupational safety classification tasks.

2.3.1. Corpus Development

This section outlines the development of a specialized corpus for finetuning and evaluating domain-adapted language models. For this purpose, two distinct datasets were employed.

The primary finetuning dataset consists of MSHA accident narratives from January 2022 to February 2025. MSHA is responsible for enforcing health and safety regulations in U.S. mining operations to prevent injuries and occupational illnesses. Under MSHA regulations, accidents are reported through Form 7000-1, completed by safety officers or supervisors, with fatal or serious injuries reported immediately and non-serious incidents within 10 days. The MSHA injury dataset contains accident reports with detailed text-based narratives along with other categorical variables including subunit, mining equipment, classification, accident type, injury source, nature of the injury, injured body parts, and degree of injury based on MSHA's classification system. Data from January 2000 to December 2021 was utilized for DAPT, while the 2022-2025 data was reserved exclusively for finetuning. This temporal separation is crucial for evaluating model performance on unseen data patterns, providing a direct comparison between PLMs and their domain-adapted counterparts.

For the finetuning of domain-adapted models on domain specific downstream task, classification of accident type, mining equipment, and degree of injury was selected due to their critical importance in occupational safety management. Accurate classification of these elements enables safety professionals to identify hazard patterns and implement targeted preventive measures. Previous research has demonstrated that understanding relationships between equipment types and accident outcomes significantly impacts safety strategy development [88] [87]. By employing both single-task and multitask learning approaches on these datasets, this study evaluates domain-adapted language model performance on unseen safety data while developing classification systems that capture the relationships between accident type, equipment, and injury severity in occupational safety contexts.

Additionally, a small Alaska mining injuries insurance claim dataset was also employed to test model performance under limited training data conditions. The non-fatal claim data were provided by the Division of Workers' Compensation for the State of Alaska during the period of 2014 to 2018. More about this data can be found elsewhere [109]. For this dataset, the models were finetuned to classify claim type and injured body part based on accident

narratives. This approach enables assessment of how domain adaptation affects performance when finetuning resources are constrained, providing insights into the models' generalizability and efficiency in resource-limited scenarios.

2.3.2. Data Preprocessing

For finetuning our domain-adapted models, the MSHA injury dataset was extracted from the MSHA database and pre-processed using the Python programming language. Pandas, NLTK, re (Regular expressions operations), and NumPy packages were used in the Python environment to process the txt file provided by the online MSHA database, as detailed in *Table 2*.

The preprocessing focused on standardizing categorical variables while preserving essential accident characteristics. The original MSHA classification system for accident type and mining equipment contains 40 and 75 classes, respectively. These were consolidated into more meaningful groups based on domain expert knowledge: 9 accident type categories and 8 mining equipment categories, as provided in *Table 4*. Additionally, *Table 4* presents the classification scheme for the degree of injury. Unlike other variables, the original MSHA degree of injury labels were deemed sufficiently granular and therefore were retained without modification for this study.

Table 4: Classification Categories for Mining Equipment, Accident Types, and Degree of Injury, presenting the consolidated grouping of original MSHA classification labels into meaningful categories for fine-tuning domain-adapted models alongside the original MSHA degree of injury classification. The mining equipment is organized into 8 functional groups, and accident types into 9 incident categories to enable more effective pattern recognition and model training.

Mining Equipment	***Grouping***
Loaders & Hauling Equipment (0)	'Load-Haul-Dump, scoop Tram, Transloader, Unitrac, s & s battery', 'Ore haulage trucks - off highway trucks', 'Ore haulage trucks - on highway trucks', 'Front-End loader, Tractor-Shovel, Payloader, Highlift, skip loader', 'Mine car, Ore or coal car, Boxcar, Hopper car', 'Mine car, Timber truck, Nipper truck', 'Shuttle car, buggy, ram car, young buggy, Teletram car', 'Gathering arm loader, Conway loader, coal loading machine', 'Boats, Barges, Other water transportation', 'Air transportation, Planes, Helicopter', 'Aerial Tram, tramway', 'Mucking machine, overshot loader, Cryderman', 'Trucks, Service truck, Utility truck, Pickup, Water truck, Fuel truck', 'Slusher, Scraper hoist, Slusher hoist'
Drilling & Breaking Equipment (1)	'Rock drill, Jackleg, Airleg, Drifter, Stoper, Buzzy, Jackhammer', 'Carriage-Mounted drills, rail, Rubber-Tired, jumbo, Air-Track drill', 'Electric drills, coal drills', 'Rock or roof bolting machine, pinning machine, Truss Bolter', 'Raise borer, raise drill', 'Raise borer, Raise drill', 'Continuous miner, Tunnel borer, Road header', 'Crusher, Breaker, Mills (ball and rod), Feeder breaker', 'Auger machine, Highwall miner, Auger highwall drill, Auger drill', 'Bench grinder, Drill press, Band/Table saw, Sandblaster', 'Machine, NEC - Wheelbarrow, Well drilling Rig, Post hole auger', 'Impactor'
Earth Moving & Ground Control (2)	'Bulldozer, Dozer, Crawler tractor, push cat', 'Road grader, Motor grader, Motor patrol, Grader, Road scraper', 'Scraper loader, Tractor scraper, Pan scraper, Elevating scraper', 'Shovel, Power shovel, Backhoe, Trackhoe, Dragline - Big Muskie', 'Hydraulic jets, Monitor', 'Gunite machine, Shotcrete ', 'Dredge', 'Raise Climber'
Processing & Separation Equipment (3)	'Classifier, Screw classifer, Spirial classifier, Cyclone classifier', 'Screen, Vibrator, Shaker, Rock screener, Scalper', 'Flotation and filters', 'Washers', 'Mill, Grinding (Rod, Ball, Autogenous, Pug, Hammer)', 'Milling machinery, Block press, Ballast machine', 'Grizzlies for coarse screening, Scalping or skimming of bulk material', 'Raw coal storage, tipple, dump bins', 'Dimension stone machinery, Gangwire saw, Guillotine, Hydrosplit', 'Rotary dump, Dump rail ', 'Chute'
Support & Utility Equipment (4)	'Hand tools (powered)', 'Hand tools (not powered)', 'Air compressor', 'Pump, Slurry pump, Sump pump', 'Fan', 'Welding machine, Torch, Cutting torch, Arc welder, Air arc', 'Track equipment, Re-railer, 335 track shifter, Track jack', 'Tamping machine', 'Rock dusting machine, Trickle duster', 'Blow pipe, Blow gun, Air lance'
Material Handling Systems (5)	'Conveyor, Belt feeder, Stage loader, Hopper shaker, Belt structure', 'Elevator, Skip, Cage, Buckets, Mancage, Slope car ', 'Packaging machine, Bagger, sewing machine, Palletizer', 'Forklift, Plant cat, Telehandler, Hyster, Lift truck, Skid steer', 'Crane, Cherry picker, Lift basket, Scissor truck, Boom truck', 'Man lift, Lift basket, Basket scaler, Self-propelled hydraulic boom', 'Tugger, Air winch, Air hoist, Air jack, Electric winch ', 'Pneumatic blasting agent loader, Driller loader, Prill loader', 'Hoist, car Dropper, Hydraulic Jack, mine car retriever, Condor'

Longwall Mining Equipment (6)	'Longwall machine', 'Longwall subparts, Duke, Dowdy jack, Ramjack, Longwall shield', 'Cutting machine, chain cutter'
Personnel & Supply Transport (7)	'Mancar, Mantrip, Personnel carrier, Porta bus, Jeep, Jitney, ATV', 'Locomotive, (motor) - rail-mounted (Battery, Steam, Electric, Air)', 'Tractor, supply car'
Accident type	***Grouping***
Without injuries (0)	"Accident type, without injuries"
Fall (1)	"Fall onto or against objects", "Fall from ladders", "Fall to the walkway or working surface", "Fall down stairs", "Fall to lower level, NEC", "Fall from machine", "Fall from scaffolds, walkways, platforms", "Fall from piled material", "Fall down raise, shaft or manway", "Fall from headframe, derrick or tower", "Fall on save level, NEC",
Caught in (2)	"Caught in, under or between a moving and a stationary object", "Caught in, under or between running or meshing objects", "Caught in, under or between NEC", "Caught in, under or between two or more moving objects", "Caught in, under or between collapsing material or buildings"
Contact with (3)	"Contact with heat", "Contact with hot objects or substances", "Contact with electrical current", "Contact with cold", "Contact with cold objects or substances"
Struck by (4)	"Struck against stationary object", "Struck against a moving object", "Struck by... NEC", "Struck by...NEC", "Struck by falling object", "Struck by flying object", "Struck by rolling or sliding object", "Struck by powered moving object", "Struck by concussion"
Over-exertion (5)	"Over-exertion in pulling or pushing objects", "Over-exertion NEC", "Over-exertion in lifting objects", "Over-exertion in wielding or throwing objects"
Exposure to Harmful Substances (6)	"Absorption of radiations, caustics, toxic and noxious substances", "Inhalation of radiations, caustics, toxic and noxious substances", "Flash burns (electric)", "Flash burns (welding)", "Ingestion of radiations, caustics, toxic and noxious substances"
Other (7)	"Drowning", "Rubbed or abraded", "Bodily reaction, NEC"
Unclassified (8)	"Unclassified, insufficient data", "NEC"
Degree of Injury	***Labels***
	'No days away from work, no restricted activity', 'Days restricted activity only', 'Days away from work only', 'Accident only', 'Days away from work & restricted activity', 'permanent total or permanent partial disability', 'Occupational illness not deg 1-6', 'All other cases (incl 1st aid)', 'Fatality', 'Injuries due to natural causes', 'Injuries involving nonemployees'

This grouping enables the models to learn more generalizable patterns rather than overfitting to highly specific categories with limited examples. Entries with missing values were removed to ensure data quality. The MSHA dataset from January 2022 to February 2025 initially contained 18,919 accident instances. After preprocessing narratives and removing entries with missing values, the dataset yielded 18,902 instances for accident type classification, 8,647 for mining equipment classification, and 18,800 for degree of injury classification. The accident narratives were retained in their original form to preserve the contextual information necessary for establishing relationships between textual descriptions and category labels. The distribution of classes for each of these variables is shown in *Figure 9*.

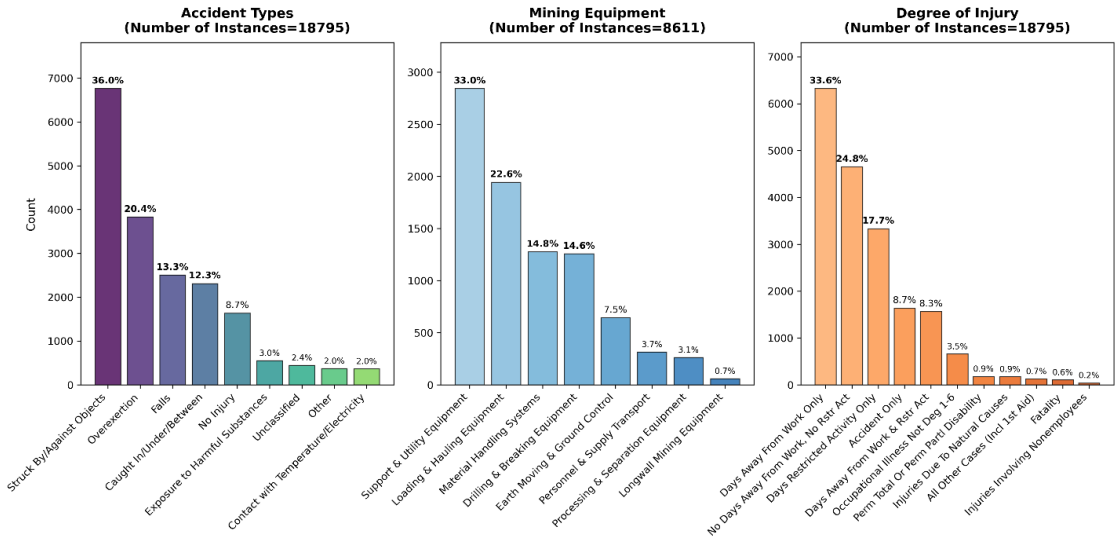


Figure 9: Distribution of Classification Categories in MSHA Finetuning Dataset, showing class distributions across three classification tasks: Accident Types (left, dominated by Struck by/against Objects at 36.0%), Mining Equipment (center, dominated by Support & Utility Equipment at 33.0%), and Degree of Injury (right, with Days Away From Work Only comprising 33.6% of instances), illustrating the class distribution and imbalance.

For the Alaska mining accident insurance claim for the non-fatal accident dataset, the original classification schema was maintained. The dataset was processed for removing missing values to ensure data integrity. From 699 initial samples, 696 remained for claim type classification (medical-only and lost time injury), and all 699 for injured body part classification (Upper body, middle body, lower body, and miscellaneous). The original categories were preserved without grouping, as they already had appropriate granularity [109]. *Figure 10* illustrates the distribution of these labels.

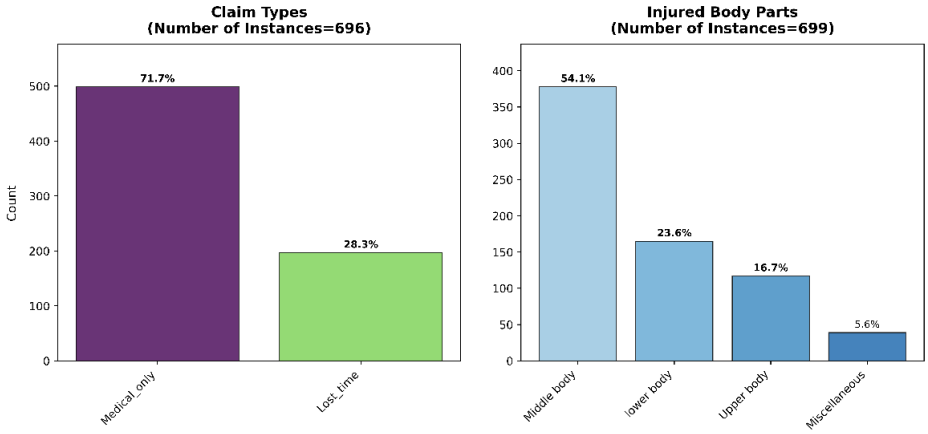


Figure 10: Distribution of Classification Categories in Alaska Mining Dataset, showing the class distribution and imbalance in both classification tasks: Claim Types (left, with Medical Only claims dominating at 71.7%) and Injured Body Parts (right, where Multiple Body injuries account for 54.1% of instances)

2.3.3. Fine-Tuning Approaches

This section presents model fine-tuning strategies to leverage domain-adapted language models for occupational safety classification tasks: single-task and multi-task learning approaches.

2.3.3.1. Single-Task Classification

The single-task learning approach addresses each classification objective independently. This approach utilizes the safetyBERT and safetyALBERT as a foundation and adds a simple classification layer. For each task, the final hidden state of the classification token ([CLS]) from the transformer encoder serves as the representation vector for the input sequence. This representation passes through a task-specific linear layer to produce logits corresponding to the target classes.

The architecture can be formalized as:

$$h = Encoder(X) \tag{10}$$

$$y = Classifier(h[CLS]) \tag{11}$$

Where *X* represents the tokenized input sequence, *h* is the sequence of hidden states produced by the encoder, *h[CLS]* is the hidden state corresponding to the classification token, and *y* represents the output logits used for classification.

During the finetuning, all transformer encoder parameters remain frozen, with only the task-specific classification head underwent training. This frozen parameter strategy treats the models as feature extractors already optimized for the occupational safety domain through the DAPT phase. Cross-entropy loss is calculated between predicted and ground truth labels, with backpropagation applied exclusively through the classification layer.

The implementation incorporates early stopping mechanisms by monitoring validation performance to prevent overfitting and ensure generalizability [110]. This single-task framework applies consistently across all classification objectives, with accident type, mining equipment, and degree of injury classifications for the MSHA dataset, and claim type and injured body part classifications for the Alaska injuries dataset.

2.3.3.2. Multi-Task Learning

The multi-task learning approach extends the single-task methodology by training multiple classification objectives. This strategy leverages the inherent relationships between different safety-related classification tasks to develop more robust and generalized representations that capture common underlying patterns, a technique that demonstrates effectiveness in various domains [111].

The multi-task implementation utilizes a shared transformer encoder with an intermediate shared representation layer, followed by parallel task-specific classification heads. This

intermediary layer with ReLU activation enhances performance by transforming the general representation into a more task-suitable form before final classification. The architecture can be formalized as:

$$h = Encoder(x) \quad (12)$$

$$y_1 = Classifier_1(h[CLS]) \quad (13)$$

$$y_2 = Classifier_2(h[CLS]) \quad (14)$$

$$y_3 = Classifier_3(h[CLS]) \quad (15)$$

Where y_1, y_2, and y_3 represent the output logits for the respective classification tasks (e.g., accident type, mining equipment, and degree of injury).

Similar to the single-task approach, all transformer encoder parameters remain frozen for the domain-adapted models (safetyBERT and safetyALBERT), while multiple task-specific classification heads undergo simultaneous training. This approach maintains the integrity of the domain-specific representations while enabling the model to capture task-specific nuances through specialized classification layers.

Loss balancing is implemented through a weighted sum of the individual task losses:

$$L = \lambda_1 L_1 + \lambda_2 L_2 + \lambda_3 L_3 \quad (16)$$

Where L_1, L_2, and L_3 represent the cross-entropy losses for each classification task, and λ_1, λ_2, λ_3 are the corresponding weights. The weights can be dynamically adjusted based on task difficulty to prevent easier tasks from dominating the training signal [112]. This study uses equal weighting with $\lambda_1 = \lambda_2 = \lambda_3 = 1$, treating all three classification tasks with equal importance. This approach was chosen to maintain balanced learning across all tasks without introducing bias toward any particular classification objective.

The multi-task framework effectively leverages the shared contextual understanding across related safety classification tasks while maintaining the unique characteristics of each specific objective. This approach is particularly advantageous in settings where limited training data is available for individual tasks, as it allows knowledge transfer across related objectives [113].

2.3.3.3. Hyperparameter Optimization

A systematic approach to hyperparameter optimization maximizes the effectiveness of domain adaptation. The methodology implements grid search with *n*-fold cross-validation to identify optimal model configurations for both single-task and multi-task learning frameworks [114]. The optimization process considers several key parameters: learning rate, batch size, weight decay for L_2 regularization, dropout rate in the classification layer, and sequence length for input narratives.

The optimization process employs a weighted F1 score as the primary evaluation metric, with accuracy, precision, and recall serving as secondary considerations, ensuring robust performance across all classes, including those with limited representation in the dataset [114]. Early stopping techniques monitor validation performance to determine the optimal training duration, with patience parameters controlling the number of epochs without improvement before termination. Section 2.4 details the experimental configuration used to implement these methodologies, including specific training parameters, architectural settings, and implementation details for both the domain adaptation and fine-tuning phases.

2.4. Experimental Setup:

2.4.1. DAPT Configuration

The DAPT of both safetyBERT and safetyALBERT was conducted for 20 epochs on the corpus of 2,458,862 documents spanning several industrial sectors, totaling approximately 342 and 350 million tokens for BERT and ALBERT, respectively. Both models employed the MLM objective with a dynamic masking strategy at a 15% masking rate. During training, both models were optimized using the AdamW optimizer with a weight decay of 0.01 and a linear learning rate decay schedule, employing initial learning rates of 1e-5 for safetyBERT and 1e-4 for safetyALBERT. To optimize memory usage during training, both models implemented identical memory efficiency techniques, including gradient accumulation and memory-aware smart batch sampling as described in Section 2.1. However, different base batch sizes were employed: 16 for safetyBERT and 32 for safetyALBERT. For both models, a dynamic batch sizing strategy was implemented that scaled the effective batch size according to sequence length categories using the same pattern for both models: 4× base batch size for very_short sequences, 2x for short, 1x for medium, 0.5x (minimum 4) for long, and 0.25x (minimum 2) for very_long sequences. This resulted in batch sizes ranging from 4-64 sequences for safetyBERT and 8-128 for safetyALBERT, enabling efficient processing of the highly variable-length sequences in the occupational safety corpus while optimizing GPU memory utilization.

The key configuration parameters for both models are presented in *Table 5*, highlighting both shared parameters and model-specific differences.

Table 5: Comparison of architectural and training parameters between safetyBERT and safetyALBERT models.

Parameter	safetyBERT	safetyALBERT
Base Architecture	bert-base-uncase	albert-base-v2
Hidden Dimension	768	768
Embedding Dimension	768	128 (factorized)
Total Parameters	~110M	~12M
Vocabulary Size	30,522	30,000

Maximum Sequence Length	512	512
Base Batch Size	16	32
Effective Batch Size	4 - 64	8 - 128
Gradient Accumulation Steps	4	4
Initial Learning Rate	1e-5	1e-4
Optimizer	AdamW	AdamW
Weight Decay	0.01	0.01
Learning Rate Schedule	Linear decay + warmup (10%)	Linear decay + warmup (10%)
Mixed Precision	No	Yes (FP16/FP32)
Maximum Training Epochs	20	20
Patience	3	3

The architectural differences between the models significantly influenced their computational requirements. The implementation of different learning rates for both models follows established practices in literature, where ALBERT-based models benefit from higher learning rates due to their parameter-efficient architecture [77], while BERT models perform optimally with lower learning rates to prevent overfitting, given their larger parameter count [39]. Similarly, mixed precision training was implemented for safetyALBERT to maintain alignment with the original implementation approaches of these models. While ALBERT was designed with computational efficiency as a primary objective, and benefits substantially from mixed precision training due to its matrix-heavy parameter-sharing approach [77].

2.4.2 Finetuning Configuration for Downstream Tasks

The downstream task evaluation employed both single-task and multi-task learning frameworks across all implemented models and datasets. We adopted a systematic fine-tuning approach for both paradigms, strategically freezing the transformer encoder parameters while training the task-specific classification layer. *Table 6* presents the hyperparameters used for fine-tuning experiments across all model architectures. For each model, we conducted a grid search over multiple parameter combinations to identify optimal configurations through 5-fold cross-validation.

Table 6: Hyperparameter configurations for fine-tuning experiments for downstream tasks across model architectures.

	Weight Decay	Patience	Learning Rate	Batch Size	Max Sequence Length	Dropout Rate	Optimizer	Loss Function
BERT/safetyBERT	0.01	5	[1e-3,1e-4, 1e-5]	[16, 32, 64]	[128, 256, 512]	[0.0, 0.1]	AdamW	Cross-Entropy
ALBERT/safetyALBERT	0.01	5	[1e-3,1e-4, 1e-5]	[16, 32, 64]	[128, 256, 512]	[0.0, 0.1]	AdamW	Cross-Entropy
Llama 3	0.01	5	[1e-3,1e-4, 1e-5]	[8, 16]	[128, 256, 512]	[0.0, 0.1]	AdamW	Cross-Entropy

For all classification tasks, including degree of injury, accident type, and mining equipment, we implemented an identical methodological approach to ensure consistency and comparability of results. Similarly, the same identical methodological approach was utilized for all classification tasks of the Alaska Injury dataset. The data partitioning process utilized

stratified sampling techniques, with training-validation-test splits of 70/15/15, resulting in 70% training, 15% validation, and 15% test data across all classification tasks. Class weighting was applied to address imbalance in the target distributions. For multi-task learning, models employed a balanced loss function combining individual task losses using configurable weight parameters of [1.0, 1.0, 1.0]. Gradient clipping (max_norm=1.0) was implemented to prevent exploding gradients.

3. Results and Discussion

This section presents the experimental results obtained from the DAPT of BERT and ALBERT models for occupational safety applications and their subsequent performance on downstream classification tasks. The results are organized into three main sections: (1) DAPT results, and (2) evaluation results assessing both intrinsic and extrinsic performance.

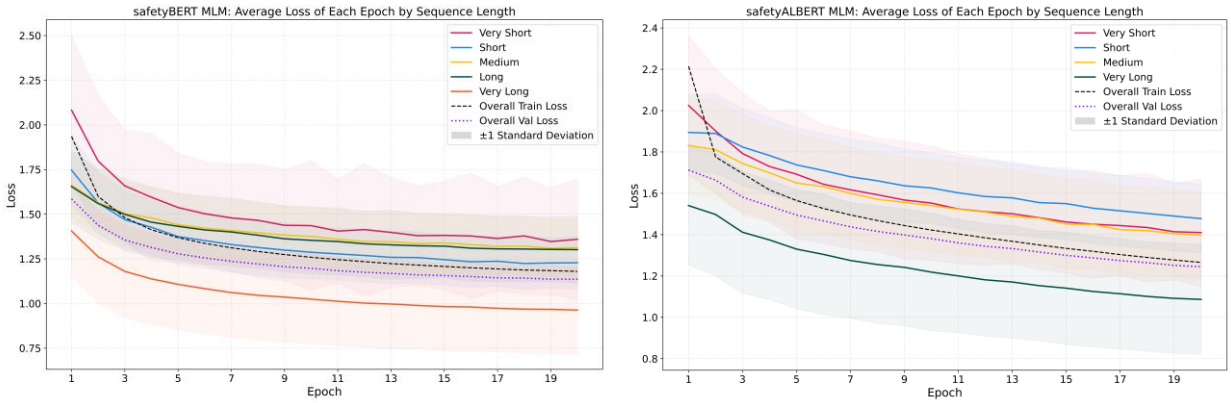
3.1. DAPT

The DAPT of BERT and ALBERT models for occupational safety applications yielded significant improvements in language modeling capabilities across both architecture variants. The learning rate strategies differed between models: safetyBERT applied a linear decay schedule from 1e-5 to 1.81e-7, while safetyALBERT used a higher initial learning rate of 1e-4 that decreased to 2.30e-6. This schedule proved effective in maintaining stable training dynamics while preventing overfitting. Throughout the training period, the safetyBERT model exhibited consistent enhancement in its language modeling capabilities, with training loss decreasing from 1.94 to 1.18 at epoch 20, representing a 39.2% reduction. Concurrently, the validation loss improved from 1.585 to 1.136, constituting a 28.3% decrease. The convergence pattern revealed the most substantial improvements during the initial five epochs, followed by more gradual improvement in subsequent iterations, indicating an effective learning trajectory with diminishing returns over time.

In comparison, the safetyALBERT model demonstrated a steeper initial learning curve, with training loss decreasing from 2.21 to 1.26 at epoch 20, representing a 42.9% reduction over the training period. The validation loss showed a corresponding improvement from 1.712 to 1.245, a 27.3% decrease. The safetyALBERT, with its factorized embedding parameterization and cross-layer parameter sharing, initially resulted in higher loss values compared to safetyBERT but ultimately achieved comparable final performance.

Analysis of model performance across different sequence lengths revealed notable variations for both architectures. The safetyBERT model exhibited superior performance on very long sequences (385-512 tokens) with an average loss of 0.963, while very short sequences (≤64 tokens) demonstrated a higher average loss of 1.360 at the end of training (epoch 20). *Figure 11a* shows consistent improvement across all sequence length categories throughout the training process, with loss values decreasing from epoch 1 to

epoch 20. Very short sequences showed the most substantial reduction in loss (34.7%), followed by very long (31.5%), short (29.7%), long (21.3%), and medium sequences (20.9%). Similarly, safetyALBERT exhibited consistent performance enhancements across all sequence length categories, with very long sequences achieving the lowest final loss (1.087), followed by medium (1.3971), short (1.478), and very short sequences (1.410) at epoch 20, with improvement of 29.4%, 23.7%, 21.9%, and 30.4%, respectively as shown in *Figure 11b*. This uniform improvement trajectory across all sequence lengths indicates the models' robust capacity to learn and generalize across varying text complexities, while consistently maintaining the relative performance advantage of longer sequences that provide richer contextual information for masked token prediction.

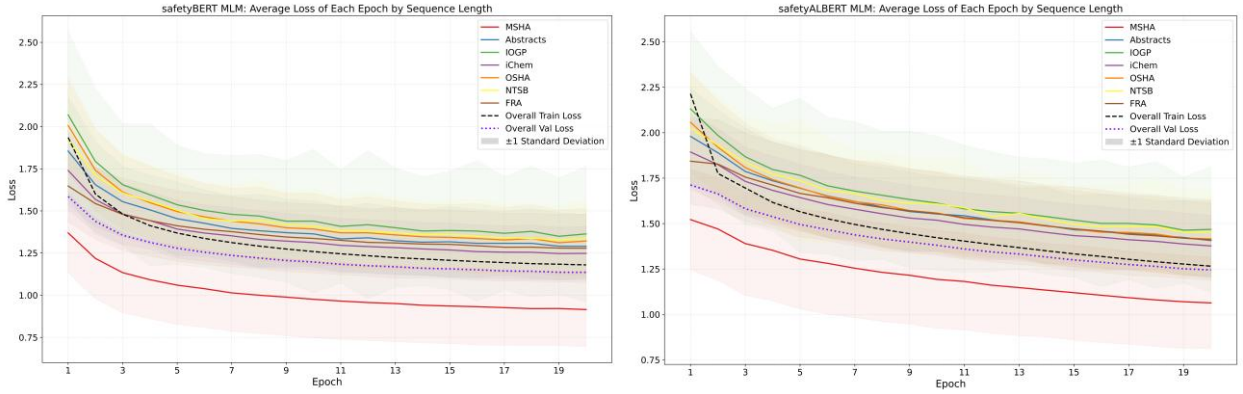


(a) (b)

Figure 11: Epoch-wise loss comparisons by sequence length for (a) safetyBERT MLM and (b) safetyALBERT MLM models. Both models show convergence over 20 epochs with different sequence length performance patterns.

The source-specific performance analysis demonstrated significant variations across all the industrial sectors represented in the training corpus. Examination of training progression shows consistent improvement across all sources. *Figure 12a* illustrates the source-specific performance where safetyBERT shows substantial improvement in training loss from epoch 1 to epoch 20, with most improvement in IOGP, showing 34.11% reduction in training loss, followed by OSHA (33.4%), MSHA (33.2%), NTSB (31.5%), academic abstracts (30.5%), iChem (26.5%), and FRA (24.3%). Similarly, safetyALBERT exhibited significant enhancements across all industrial sectors, with most improvement in IOGP, showing 31.1% reduction in training loss, followed by MSHA (30.2%), NTSB (30%), OSHA (29.4%), Academic abstracts (28.8%), iChem (25.8%), and FRA (25.3%) as shown in *Figure 12b*. These results suggest highly effective adaptation to safety-specific terminologies and narrative structures. The consistent cross-source improvement pattern, coupled with the persistent performance differential between sources, indicates the models' capacity to learn domain-specific language patterns while reflecting inherent variations in linguistic complexity across diverse occupational safety domains.

Notably, these results validate the effectiveness of our weighted sampling strategy for addressing source domain imbalance. Despite the significant corpus disparity (OSHA at 41.5% vs. IOGP and IChemE at just 0.3% and 0.6%), underrepresented sources like IOGP showed better relative improvement in both models. This balanced learning across sources demonstrates that the inverse-frequency weighting successfully prevented dominant sources from overwhelming the training signal while ensuring that specialized terminology from minority sources was adequately learned during domain adaptation.



(a) (b)

Figure 12: MLM training loss by dataset source for (a) safetyBERT and (b) safetyALBERT models. Both architectures show consistent loss reduction across 20 epochs for MSHA, abstracts, DOI, patents, OSHA, and NTSB datasets, with MSHA data yielding the lowest overall loss.

The total training duration for safetyBERT was approximately 6 days and 19 hours (163 hours), while safetyALBERT required approximately 4 days and 7 hours (103 hours) to complete all 20 epochs on identical NVIDIA A100-40GB GPU hardware. When accounting for the difference in epoch count, safetyALBERT demonstrated greater training efficiency, requiring approximately 5.1 hours per epoch compared to safetyBERT's 8.2 hours per epoch - a 37.8% reduction in per-epoch training time despite processing an identical corpus. This efficiency gain directly reflects ALBERT's architectural optimizations for computational performance.

Both models demonstrated effective domain adaptation with progressive improvement throughout training, with minimal signs of overfitting as evidenced by the continued convergence of training and validation losses. The resulting language models (safetyBERT and safetyALBERT) provide robust foundations for downstream occupational safety applications. The comparatively strong performance of safetyALBERT, despite its significantly reduced parameter count, highlights the effectiveness of parameter-efficient architectures for domain-specific language modeling in the specialized occupational safety domain. These findings establish a foundation for evaluating the models' domain-specific language understanding, which is further examined through intrinsic and extrinsic evaluations in the subsequent sections.

3.2. Model Evaluation

3.2.1. Intrinsic Evaluation:

Intrinsic evaluation through PPPL on held-out occupational safety data demonstrated substantial improvements compared to pretrained counterparts. The evaluation was conducted on a dataset of 18,907 accident narratives from MSHA, comprising approximately 863,138 tokens for ALBERT-based models and 841,919 tokens for BERT-based models. safetyBERT model achieved a PPPL of 3.04, representing a 76.9% reduction from the pretrained BERT baseline (PPPL: 13.2). Similarly, safetyALBERT resulted in a PPPL of 3.4, constituting a 90.3% reduction from the pretrained ALBERT baseline (PPPL: 35.06).

Comparatively, safetyBERT achieved a lower PPPL than safetyALBERT (3.04 vs 3.4), showing superior performance of safetyBERT over safetyALBERT in the safety domain. This 10.6% difference could be attributed to BERT's larger parameter count and consequent higher capacity for domain-specific knowledge representation. However, when considering the efficiency-performance trade-off, safetyALBERT's performance remains particularly impressive given its parameter-efficient architecture.

These results demonstrate that both domain-adapted models have developed significantly enhanced representations of occupational safety language compared to their general-domain counterparts, demonstrating the potential of continual training to enhance the PLMs ability to capture domain-specific linguistic nuances. The relative improvement from domain adaptation is higher for ALBERT than for BERT, suggesting that smaller, parameter-efficient models may benefit even more from domain-specific adaptation. The following section will examine how these intrinsic language modeling improvements translate to extrinsic performance on specific occupational safety applications.

3.2.2. Downstream Task Evaluation

3.2.2.1. Single-Task Classification - MSHA Dataset

The single-task classification experiments on the held-out MSHA dataset investigated model performance on occupational safety-related downstream tasks: degree of injury classification, accident type classification, and equipment type classification. Each task presents unique challenges, ranging from highly imbalanced class distributions to domain-specific terminology that requires specialized understanding. A grid search with 5-fold cross-validation was conducted to identify optimal configurations for each model and task. *Table 7* presents the optimal hyperparameters and corresponding validation performance across the three classification tasks.

Table 7: Optimal hyperparameters and performance metrics for single-task classification on MSHA safety datasets. Results compare base and safety-tuned models across three classification tasks, showing consistent improvements from domain-specific pretraining.

Model	Best Batch Size			Best Max Length			Best Avg Validation F1		
	Degree of Injury	Accident Type	Equipment Type	Degree of Injury	Accident Type	Equipment Type	Degree of Injury	Accident Type	Equipment Type
BERT (base)	32	64	32	512	512	512	0.4156 ± 0.0273	0.4643 ± 0.0289	0.4025 ± 0.0236
safetyBERT	**32**	**32**	**32**	**512**	**512**	**512**	**0.5237 ± 0.0124**	**0.7674 ± 0.0058**	**0.7136 ± 0.0153**
ALBERT (base)	64	32	32	256	256	256	0.3427 ± 0.0340	0.3688 ± 0.0220	0.3193 ± 0.0455
safetyALBERT	**64**	**32**	**32**	**256**	**256**	**256**	**0.4969 ± 0.0111**	**0.6775 ± 0.0091**	**0.6950 ± 0.0137**
Llama 3	**16**	**16**	**8**	**512**	**512**	**512**	**0.4935 ± 0.0122**	**0.7043 ± 0.0027**	**0.6203 ± 0.0182**

The hyperparameter search revealed consistent learning rate preferences across all models and tasks, with 1e-3 proving optimal in nearly all cases (with Llama 3 on Equipment Type classification being the sole exception at 1e-4). All models maintained consistent weight decay (0.01) and employed early stopping with a patience of 5 epochs. Architectural differences emerged in sequence length preferences. Pretrained BERT models and Llama 3 generally performed optimally with longer sequences (512 tokens). Conversely, ALBERT-based models typically achieved superior performance with shorter sequences (256 tokens).

The average validation F1 scores across all folds demonstrated substantial performance improvements from domain adaptation across all tasks and provided greater cross-validation stability, evidenced by consistently lower standard deviations across all classification tasks. This suggests that domain adaptation not only improved performance but also enhanced model robustness.

3.2.2.1.1. Performance Analysis

The performance metrics for all the classification tasks across all models are summarized in *Table 8*, highlighting test accuracy, macro-averaged precision, recall, and F1-scores. *Figure 13* provides a performance comparison of models across all classification tasks on test data.

Table 8: Performance metrics for single-task classification across model architectures on test data. Results compare base and safety-tuned models on three MSHA classification tasks (Degree of Injury, Accident Type, and Equipment Type). SafetyBERT consistently outperformed all models across all metrics, with domain-adapted models showing substantial improvements over their general-domain counterparts across all tasks.

Model	Accuracy			Macro Avg Precision			Macro Avg Recall			Macro Avg F1			Weighted Avg F1		
	Degree of Injury	Accident Type	Equipment Type	Degree of Injury	Accident Type	Equipment Type	Degree of Injury	Accident Type	Equipment Type	Degree of Injury	Accident Type	Equipment Type	Degree of Injury	Accident Type	Equipment Type
BERT (base)	0.45	0.52	0.50	0.25	0.54	0.26	0.22	0.29	0.26	0.21	0.28	0.23	0.42	0.466	0.4329

safetyBERT	**0.56**	**0.78**	**0.75**	**0.53**	**0.81**	**0.79**	**0.46**	**0.64**	**0.55**	**0.47**	**0.66**	**0.60**	**0.52**	**0.768**	**0.7275**
ALBERT (base)	0.417	0.45	0.42	0.21	0.33	0.21	0.18	0.22	0.19	0.17	0.21	0.17	0.35	0.385	0.3314
safetyALBERT	**0.54**	**0.70**	**0.73**	**0.42**	**0.64**	**0.77**	**0.37**	**0.53**	**0.52**	**0.39**	**0.56**	**0.56**	**0.506**	**0.681**	**0.7166**
Llama 3	**0.53**	**0.74**	**0.68**	**0.34**	**0.68**	**0.55**	**0.30**	**0.53**	**0.43**	**0.31**	**0.57**	**0.44**	**0.51**	**0.721**	**0.6517**

The experimental results demonstrate several significant findings. The Degree of injury classification task proved most challenging overall, with lower performance metrics across all models compared to other tasks. Domain-adapted models demonstrated substantial improvements over their general-domain counterparts, with safetyBERT achieving an 11% improvement over base BERT, and safetyALBERT showing a 12% improvement over base ALBERT. The safetyBERT model demonstrated the strongest overall performance with the highest test accuracy (56%) and macro-average metrics (precision: 0.53, recall: 0.46, F1: 0.47).

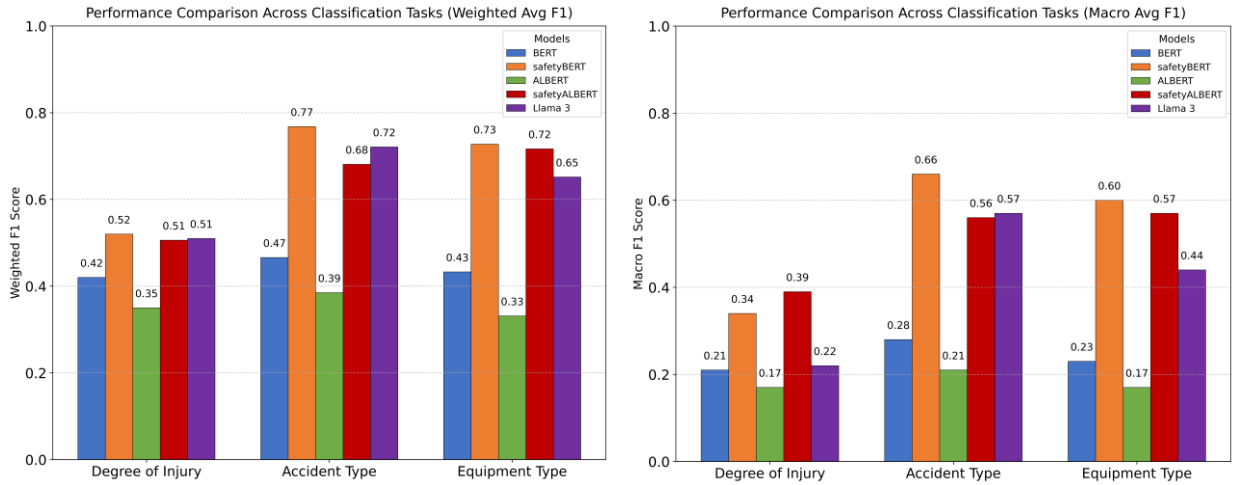


Figure 13: Model performance comparison across classification tasks using (a) weighted average F1 and (b) macro average F1 scores. SafetyBERT consistently outperforms all other models across all tasks, with the most significant improvements observed in Accident Type and Equipment Type classification. pretrained models show markedly lower performance compared to their safety-domain counterparts

Figure 14 shows class-specific performance analysis, revealing significant variations across the degree of injury categories. For the "Accident only" class, safetyBERT achieved an exceptional performance with precision and recall scores above 0.90. Similarly, for "Days Away from Work Only" and "No Days Away from Work, No Restricted Activity" categories, both safetyBERT and safetyALBERT demonstrated superior precision-recall trade-offs compared to their general-domain counterparts. Most notably, domain-adapted models showed greater capability to identify rare injury categories. SafetyBERT and safetyALBERT were able to classify a few instances of "Fatality" and "Injuries due to natural causes", while pretrained BERT and ALBERT completely failed to identify these categories. All models faced challenges with certain injury classes, particularly "All other cases (Including First AID),"

"Days away from work & restricted activity," "Injuries involving non-employees," and "Permanent total or permanent partial disability." These categories had either limited representation in the training data or shared similar contextual features in their narrative descriptions with more common classes. The "Permanent Total or Permanent Partial Disability" class proved especially challenging due to its rarity and potential overlap with other severity descriptions in the narrative text. Llama 3 showed interesting class-specific behaviors, achieving strong performance on the "Accident only" category but struggling with balanced classification across other injury types, demonstrating a tendency to favor more common injury categories at the expense of rarer categories.

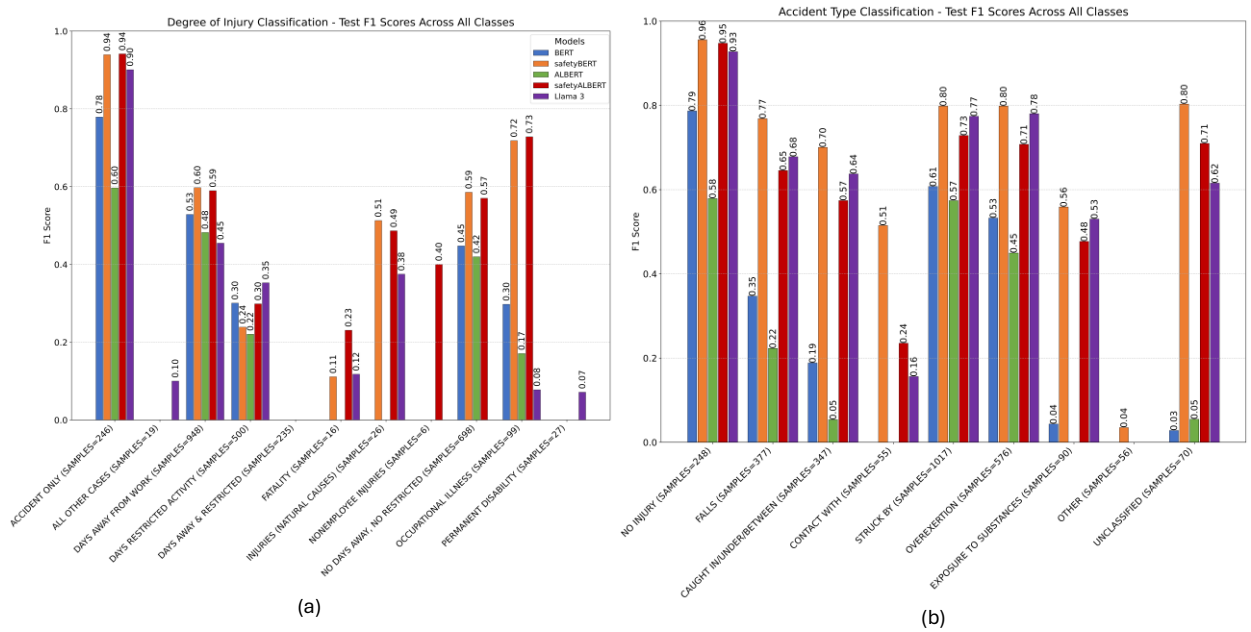


Figure 14: Class-wise F1 score comparison across models for (a) Degree of Injury and (b) Accident Type classification tasks. Safety-tuned models consistently outperform their base counterparts across most classes, with safetyBERT showing the highest performance on the majority of injury types and accident categories. Notable performance gaps appear in some rare classes (e.g., "Permanent Disability").

Similarly, the accident type classification task also showed substantial performance improvements from domain adaptation. SafetyBERT achieved a 26% improvement over pretrained BERT, while safetyALBERT demonstrated a 25% improvement over pretrained ALBERT. These improvements were consistent across all evaluation metrics, with particularly notable gains in macro-average metrics, indicating more balanced performance across accident categories. Llama 3 also demonstrated strong performance on this task (74% accuracy, 0.72 weighted F1).

Class-specific analysis of accident type classification revealed diverse performance patterns across accident types, as shown in *Figure 14*. For "Accident type, without injuries" accidents, safetyBERT achieved better performance with an F1 score of 0.96, outperforming pretrained BERT (0.79). Llama 3 also performed better on this category with an F1 score of

0.93. Similarly, for "Struck by/against objects" accidents, which represented the largest category in the dataset, safetyBERT, safetyALBERT, and Llama 3 all demonstrated strong performance with F1 scores of 0.80, 0.73, and 0.77, respectively, compared to pretrained BERT (0.61). The domain-adapted models and Llama 3 demonstrated superior ability to correctly classify challenging accident types. For "Caught in/under/between" incidents, which pretrained BERT and ALBERT struggled to identify (0.19 and 0.05, respectively), safetyBERT achieved an F1 score of 0.70, safetyALBERT reached 0.57, and Llama 3 attained 0.64. Similarly, for rare "Contact with" accidents, which PLMs completely failed to identify (0.00), domain-adapted models demonstrated meaningful classification capability (safetyBERT: 0.51, safetyALBERT: 0.24), though Llama 3 struggled more with this rare category (0.16).

The equipment type classification task showed similar improvements from domain adaptation. SafetyBERT achieved a 25% improvement over pretrained BERT, while safetyALBERT demonstrated an even more pronounced 31% improvement over pretrained ALBERT. These improvements were consistent across all evaluation metrics, with notable gains in macro-averaged metrics, indicating more balanced performance across equipment categories. The safetyBERT model demonstrated the strongest overall performance with a weighted F1 score of 0.7275, outperforming safetyALBERT (0.7166).

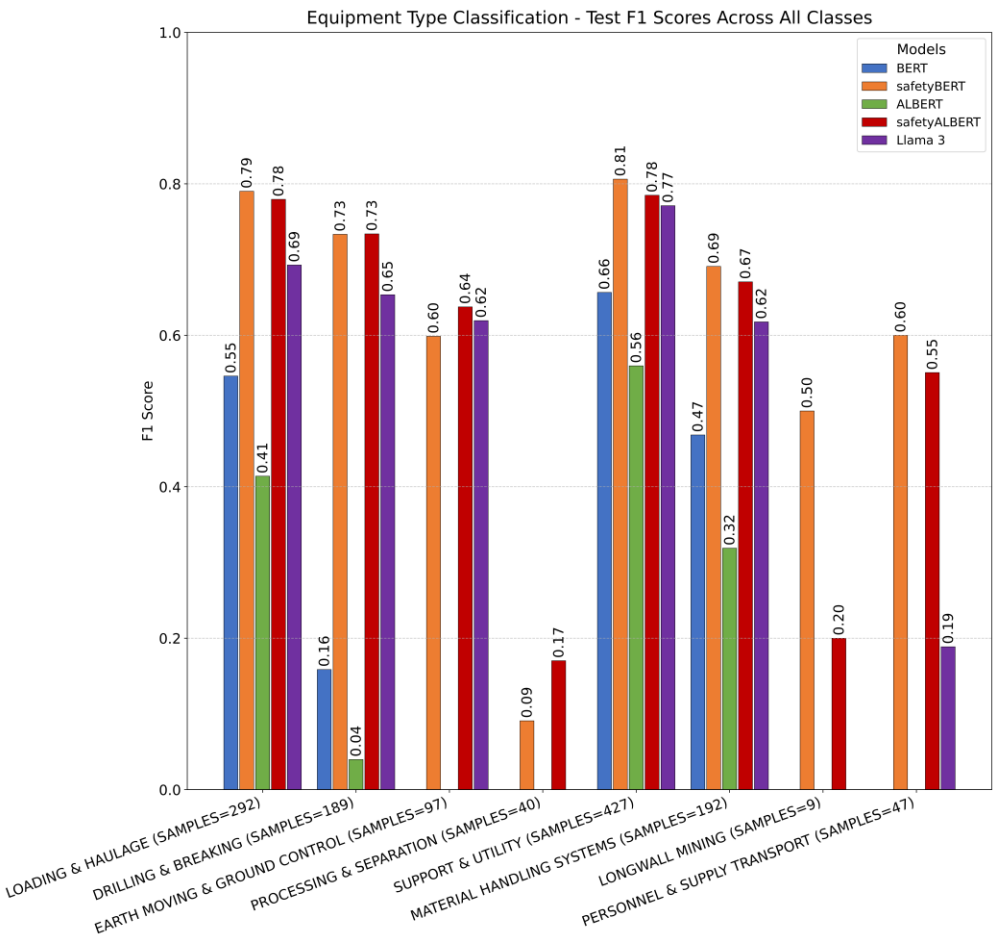


Figure 15: Class-wise F1 score comparison across models for Equipment Type classification. SafetyBERT demonstrates superior performance across all equipment categories. Base models consistently underperform on low-sample classes, while safety-domain variants maintain reasonable effectiveness even with limited training examples.

Figure 15 shows class-specific analysis revealing significant variations across equipment types. For "Loading & Hauling Equipment", safetyBERT and safetyALBERT achieved better performance with F1 scores of 0.79 and 0.77, respectively, compared to pretrained BERT (0.55) and Llama 3 (0.69). Similar improvements were observed for "Support & Utility Equipment", where safetyBERT, safetyALBERT, and Llama 3 achieved F1 scores of 0.81, 0.78, and 0.77, respectively, compared to pretrained BERT (0.66). Most notably, domain-adapted models demonstrated superior capability to classify rare equipment types. For "Earth Moving & Ground Control", which pretrained BERT and ALBERT completely failed to identify (F1 score of 0.00), safetyBERT and safetyALBERT achieved F1 scores of 0.60 and 0.64, respectively, with Llama 3 achieving 0.62. Similarly, for "Personnel & Supply Transport", domain-adapted models achieved meaningful classification capability (safetyBERT: 0.60, safetyALBERT: 0.55), while Llama 3 struggled with this rare category (F1 score of 0.19). The most challenging categories for all models were "Processing & Separation Equipment" and "Longwall Mining Equipment", which had the smallest representation in the dataset (40 and

9 instances, respectively). Even here, domain-adapted models showed measurable improvement, with safetyBERT achieving F1 scores of 0.1 and 0.5, while safetyALBERT achieved F1 scores of 0.17 and 0.2 - compared to pretrained BERT, ALBERT, and Llama 3, which completely failed to identify these classes (F1 score of 0).

These results demonstrate that domain-adapted models consistently outperformed their general-domain counterparts across all classification tasks. These safety domain models were also able to classify infrequent categories, such as “Fatality” to some extent that general models completely failed to classify, confirming that domain adaptation significantly enhances occupational safety applications.

3.2.2.2. Single-Task Classification Results - Alaska insurance claim Dataset

The Alaska insurance claim from mining Injury dataset provided an opportunity to evaluate model performance under limited data conditions across two classification tasks: claim type and injured body part classification. To identify optimal configurations for each model architecture, a systematic grid search was conducted with 5-fold cross-validation. Given the size of the dataset and the class imbalance, the search space encompassed learning rates [1e-3, 1e-4, 1e-5], batch sizes [4, 8] for BERT and ALBERT models, and maximum sequence lengths [256, 512]. All models employed consistent weight decay values (0.01) across architectures, used weighted F1 score as the primary optimization metric, and implemented early stopping with a patience of 5 epochs. *Table 9* presents the hyperparameter optimization results for all models.

Table 9: Optimal hyperparameters and validation performance for Alaska Injury dataset classification tasks.

Model	Best Learning Rate	Weight Decay	Best Batch Size		Best Max Length		Best Avg Validation F1	
			Claim Type	Injured Part	Claim Type	Injured Part	Claim Type	Injured Part
BERT (base)	0.001	0.01	4	64	256	512	0.5720 ± 0.0618	0.4643 ± 0.0289
safetyBERT	**0.001**	**0.01**	**4**	**32**	**256**	**512**	**0.6321 ± 0.0254**	**0.7674 ± 0.0058**
ALBERT (base)	0.0001	0.01	8	32	256	256	0.5562 ± 0.0198	0.3688 ± 0.0220
safetyALBERT	**0.001**	**0.01**	**4**	**32**	**256**	**256**	**0.6038 ± 0.0565**	**0.6775 ± 0.0091**
Llama 3	**0.0001**	**0.01**	**8**	**16**	**512**	**512**	**0.6157 ± 0.0441**	**0.7043 ± 0.0027**

3.2.2.2.1. Performance Analysis

The performance metrics for all the classification tasks across all models are summarized in

Table 10, highlighting test accuracy, macro-averaged precision, recall, and F1-scores. *Figure 16* provides a performance comparison of models across all classification tasks.

Table 10: Performance metrics for Alaska Injury dataset classification across model architectures on test data. Results demonstrate domain-adapted models' superiority, with safetyALBERT achieving the highest accuracy on Claim Type classification and safetyBERT on Injured Part classification. While Llama 3 achieved the highest

Claim Type accuracy, its low macro-average metrics indicate class imbalance issues not present in safety-domain models.

Model	Accuracy		Macro Avg Precision		Macro Avg Recall		Macro Avg F1		Weighted Avg F1	
	Claim Type	Injured Part	Claim Type	Injured Part	Claim Type	Injured Part	Claim Type	Injured Part	Claim Type	Injured Part
BERT (base)	0.5577	0.4423	0.5258	0.3071	0.5306	0.2633	0.5175	0.2244	0.5764	0.3959
safetyBERT	**0.5962**	**0.7692**	**0.5576**	**0.7014**	**0.5676**	**0.6807**	**0.5539**	**0.6852**	**0.6120**	**0.7652**
ALBERT (base)	0.5673	0.3750	0.4418	0.1876	0.4482	0.2272	0.4439	0.2018	0.5547	0.3520
safetyALBERT	**0.6442**	**0.6827**	**0.5553**	**0.6070**	**0.5518**	**0.6916**	**0.5529**	**0.6349**	**0.6384**	**0.6903**
Llama 3	**0.6827**	**0.5385**	**0.5246**	**0.3111**	**0.5095**	**0.3042**	**0.4793**	**0.2458**	**0.6170**	**0.4822**

The claim type classification task showed moderate improvements from domain adaptation. SafetyBERT achieved a 6.2% relative improvement in weighted F1 over base pretrained BERT, while safetyALBERT demonstrated a more substantial 15.1% improvement over ALBERT. Notably, the macro-average F1 scores for domain-adapted models were substantially higher than both their general-domain counterparts and Llama 3, despite Llama 3 achieving the highest accuracy. *Figure 17*a provides the class-specific performance of all the models on test data. This analysis revealed that Llama 3 exhibited extreme class imbalance, correctly identifying only 10% of lost time claims while achieving the highest medical only recall (91.9%) among all models. This behavior highlights the potential challenges of applying larger language models to smaller specialized datasets with class imbalance without domain-specific adaptation.

Figure 16: Model performance comparison on Alaska Injury dataset using (a) weighted average F1 and (b) macro average F1 scores. SafetyBERT demonstrates superior performance on both tasks, particularly on Injured Body Part classification (0.76 weighted F1), where domain adaptation yields the most significant improvements. The substantial difference between weighted and macro F1 scores for Llama 3 confirms its class imbalance issues.

The injured body part classification task demonstrated comparatively better performance improvement from domain adaptation across all experiments, as depicted in *Figure 17b*.

Based on the test results in Table 6, safetyBERT achieved a 93.3% relative improvement in weighted F1 score over pretrained BERT, while safetyALBERT showed a 96.1% improvement over ALBERT, and both models demonstrated significantly better performance in macro-average F1 scores. These improvements in macro F1 scores reflect the substantially more balanced performance of domain-adapted models. This balanced performance demonstrates a contrast to general-domain models, which showed significantly lower macro F1 scores. However, Llama 3 showed moderately better performance compared to general-domain BERT and ALBERT.

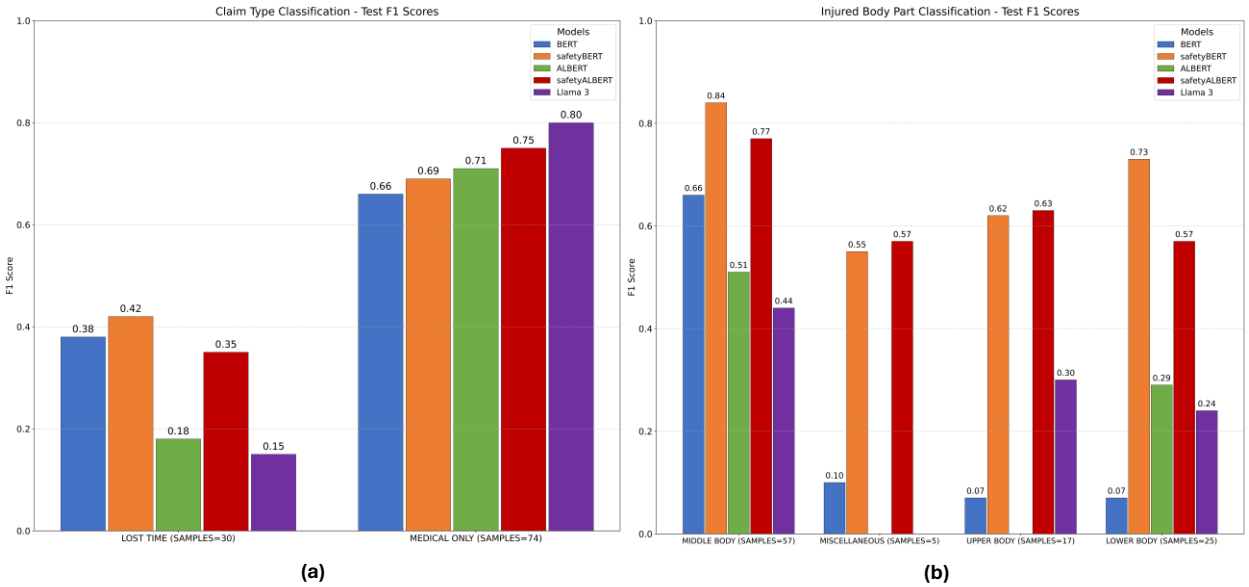


Figure 17: Class-wise F1 score comparison across models for (a) Claim Type and (b) Injured Body Part classification tasks. In Claim Type classification, Llama 3 shows extreme class bias toward Medical Only claims (0.80) while performing poorly on Lost Time claims (0.15). For Injured Body Part classification, safetyBERT consistently achieves the highest F1 scores across all classes.

These results illustrate that domain adaptation provides significant advantages even in resource-constrained environments, with specialized models achieving more balanced performance across all categories compared to both general models and Llama 3, which struggled with class imbalance despite achieving high accuracy on majority classes.

3.2.2.3. Multi-task Classification

The multi-task learning framework was evaluated across all model architectures on the MSHA dataset, encompassing three distinct classification tasks: degree of injury, accident type, and mining equipment. A comprehensive grid search was conducted to identify optimal configurations for each model architecture in the multi-task learning setting. The search space examined different parameter combinations with learning rates [1e-3, 1e-4, 1e-5], batch sizes [16, 32, 64], maximum sequence lengths [256, 512], and weight decay of 0.01. Task weights were maintained at [1.0, 1.0, 1.0] across all experiments to provide equal

importance to each classification objective. The optimal configurations demonstrated consistent learning rate preferences across all transformer-based architectures (0.001), while showing some variation in batch size and sequence length parameters. *Table 11* summarizes the optimal hyperparameters and corresponding performance metrics for each model architecture in the multi-task learning framework.

Table 11: Optimal hyperparameters and performance metrics for multi-task learning on the MSHA dataset.

Model	Batch Size	Max Length	Learning Rate	Weight Decay	Best Avg CV F1 (Combined)
BERT	64	512	0.001	0.01	0.4981 ± 0.0113
safetyBERT	32	512	0.001	0.01	0.7018 ± 0.0051
ALBERT	32	256	0.001	0.01	0.5504 ± 0.0047
safetyALBERT	32	512	0.001	0.01	0.6857 ± 0.0014
Llama 3	32	512	0.001	0.01	0.5834 ± 0.0021

3.2.2.3.1. Performance Analysis

Table 12 presents comparative performance metrics across all evaluated models for the multi-task classification approach.

Table 12: Multi-task classification performance across model architectures on the MSHA test dataset. safetyBERT and safetyALBERT demonstrate superior performance across all tasks.

Model	DEGREE_INJURY		ACCIDENT_TYPE		MINING_EQUIP		Combined Test F1
	Accuracy	Macro Avg F1	Accuracy	Macro Avg F1	Accuracy	Macro Avg F1	
BERT	0.5256	0.2075	0.6186	0.3175	0.5581	0.2629	0.5085
safetyBERT	**0.5698**	**0.3230**	**0.8419**	**0.6793**	**0.7674**	**0.6455**	**0.7041**
ALBERT	0.5116	0.2410	0.6651	0.3659	0.6209	0.3556	0.5591
safetyALBERT	**0.5698**	**0.2713**	**0.7953**	**0.6329**	**0.8047**	**0.7269**	**0.7056**
Llama3	**0.4916**	**0.2316**	**0.7728**	**0.5232**	**0.5826**	**0.4431**	**0.5989**

The performance metrics reveal the differences between domain-adapted and general-domain models in a multitask learning environment. The safetyBERT and safetyALBERT models demonstrated better performance, achieving a 41.1% and 26.2% relative improvement, respectively, over pretrained BERT and ALBERT in the combined test F1 score. *Figure 18* shows the individual tasks' performance of all models, where the most pronounced improvements were observed in the accident type and mining equipment classifications. For Accident Type, safetyBERT and safetyALBERT achieved an F1 score of 0.84 and 0.79, compared to BERT and ALBERT achieving an F1 score of 0.57 and 0.63, respectively. Similarly, for the mining equipment task, safetyBERT and safetyALBERT achieved an F-1 score of 0.75 and safetyALBERT achieved 0.8, respectively. Comparatively, the Llama 3 model also outperformed both general-domain BERT and ALBERT.

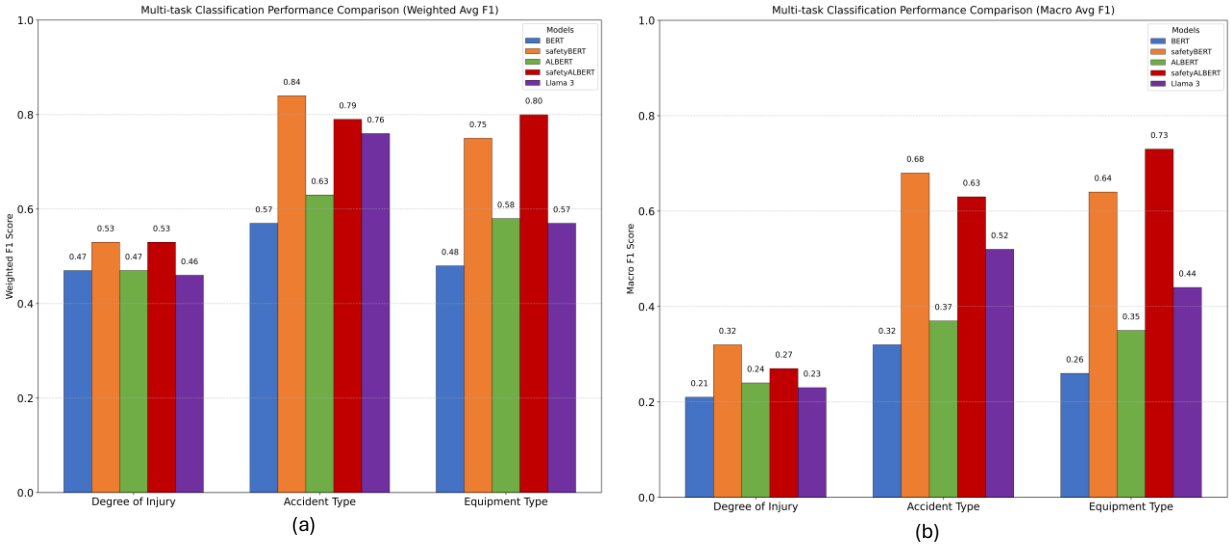


Figure 18: Multi-task learning performance comparison using (a) weighted average F1 and (b) macro average F1 scores across classification tasks. Domain-adapted models consistently outperform their base counterparts across all tasks, with the most significant improvements observed in Accident Type and Equipment Type classification.

Further examination of task-specific metrics shows that for the degree of injury classification, all models struggled with rare injury categories. The safetyBERT model demonstrated the most balanced performance, correctly identifying 4 of 11 injury classes with F1 scores above 0.40, compared to the pretrained BERT and Llama 3. Notably, only domain-adapted models successfully classified the critical "Fatality" category with a meaningful F1 score of 0.67 and 0.55 for safetyBERT and safetyALBERT, respectively. Similarly, in the accident type classification, domain-adapted models demonstrated more balanced performance across all categories. SafetyBERT and safetyALBERT classified 7 out of 9 accident types with F1 scores above 0.50, compared to pretrained BERT and ALBERT, which classified 3 and 4 classes, respectively. The mining equipment classification revealed significant improvements from domain adaptation, where both safetyBERT and safetyALBERT demonstrated more balanced performance across all categories, however, safetyALBERT showed superior performance and classified 8 out of 8 equipment categories with F1 scores above 0.48, compared to pretrained BERT and ALBERT, which classified 3 and 4 classes, respectively.

The comparative analysis between single-task and multi-task approaches reveals that multi-task learning particularly benefits certain safety dimensions, with accident type and equipment type classifications showing substantial improvements when learned jointly. Domain-adapted models demonstrate notable benefits in the multi-task setting, with safetyBERT achieving an 84% weighted F1 score for accident classification compared to 77% in single-task learning, while safetyALBERT shows remarkable improvement in equipment type classification - its macro F1 score increasing from 0.56 to 0.73 in the multi-

task framework, outperforming the larger safetyBERT model (0.64). Most importantly, multi-task learning enhanced balanced performance across all safety categories, including rare but critical events, providing a more comprehensive understanding of safety incidents.

Conclusion

This study demonstrates the effectiveness of DAPT for enhancing pretrained language models in the occupational safety domain. By continually pretraining BERT and ALBERT architectures on a diverse, multi-source corpus of 2.4 million safety documents spanning several industrial sectors and academic abstracts, we developed safetyBERT and safetyALBERT, domain-specialized models. The comprehensive investigation in this study establishes safetyBERT and safetyALBERT as highly effective models for occupational safety applications across all evaluation dimensions. The MLM training demonstrated significant improvements in both models' language modeling capabilities, with safetyBERT achieving a 39.2% and 28.3% reduction in training loss and validation loss, respectively. Similarly, safetyALBERT shows an even more substantial 42.9% and 27.3% reduction in training and validation loss, respectively. The source-specific analysis also revealed consistent improvements across all industrial sectors for safety domain models, with the weighted sampling approach appearing to be effective in addressing source domain imbalance. Despite OSHA narratives comprising 41.5% of our corpus while IOGP and IChemE contributed only 0.3% and 0.6% respectively, both models exhibited consistent improvement across all sources. This balanced learning was evidenced by differential but systematic improvements: safetyBERT showed the greatest reduction for IOGP (34.11%), followed by OSHA (33.4%) and MSHA (33.2%), while safetyALBERT demonstrated similar patterns with 31.1%, 29.4%, and 30.2% respectively. These results validate our weighted sampling strategy, ensuring neither model is overfitted to dominant sources while still capturing specialized terminology from underrepresented sectors.

The intrinsic evaluation through PPPL metrics quantified these language modeling improvements, with safetyBERT achieving a 76.9% reduction (PPPL: 3.04 vs. 13.2) and safetyALBERT demonstrates an even more remarkable 90.3% reduction (PPPL: 3.4 vs. 35.06) compared to their pretrained counterparts. This substantial difference in relative improvement suggests that parameter-efficient architecture may benefit even more substantially from domain-specific adaptation. These intrinsic improvements translated directly to superior extrinsic performance across all classification tasks. SafetyBERT consistently outperformed base BERT by 11-26% over base BERT across all tasks (11% for degree of injury, 26% for accident type, and 25% for equipment type), while safetyALBERT demonstrated even more substantial improvements of 12-31% over base ALBERT (12% for degree of injury, 25% for accident type, and 31% for equipment type). Most crucially, both

safety-domain models showed improvement in the classification of critical and rare categories with minimal training examples, which general trained models completely failed to detect.

In data-constrained and class-imbalanced settings, both safety-domain models maintained their performance advantages, with safetyBERT and safetyALBERT performing significantly better for all tasks when compared to their base models in the Alaska insurance claim from mining injury dataset. Their balanced classification capabilities across all the tasks (claim type and injured body part) and categories, including rare events, substantially outperformed both general models and the larger Llama 3 architecture. Our comparative analysis with the substantially larger Llama 3.1-8B model further reinforced the value of domain-specific adaptation. Despite its advanced architecture, Llama 3 consistently underperformed on specialized safety tasks as compared to safety domain models, particularly struggling with balanced classification across categories. This finding establishes that domain adaptation is beneficial for optimal performance in specialized technical domains.

The multi-task learning framework revealed additional integrative advantages across classification objectives. SafetyBERT's accident type classification performance increased from 77% to 84% weighted F1 when transitioning from single-task to multi-task training, while safetyALBERT exhibited remarkable enhancement in equipment classification, with macro F1 scores rising from 0.56 to 0.73 - significantly outperforming safetyBERT's 0.64. Notably, both domain-adapted models demonstrated robust capability in classifying critical rare categories such as "Fatality" (safetyBERT: 0.67 F1, safetyALBERT: 0.55 F1), while such classification performance remained elusive for general models in single-task configurations. The joint learning architecture, built upon domain-specialized foundations, effectively captured interdependencies between safety factors, fostering more balanced category-wise performance and substantiating the superiority of domain-adapted models in multi-task frameworks and in modeling complex accident characteristic relationships.

Overall, this research presents a scalable and resource-efficient strategy for domain adaptation in occupational safety NLP applications. By showing that continual pretraining on a curated multi-source corpus leads to measurable gains in both language modeling and downstream task performance, this work contributes a robust methodological foundation for future efforts in safety-focused AI. These findings have meaningful implications for occupational safety management, particularly in enabling automated analysis of unstructured incident narratives for proactive hazard identification and risk mitigation. Future research may extend this work through exploration of cross-domain transferability, zero-shot or few-shot adaptation in low-resource scenarios, and integration of structured safety knowledge into PLMs for even deeper contextual reasoning.

Acknowledgements

We gratefully acknowledge the National Institute for Occupational Safety and Health (NIOSH) for their partial support of this research under Contract Number 75D30121C12375. The second author also acknowledges the support provided by the Witte Family Faculty Fellow funds. The authors also acknowledge the use of AI language models in improving the clarity and readability of this manuscript's technical writing, while maintaining the complete scientific integrity of the content.

This research was made possible in part by computing resources provided by JetStream2 under the National Science Foundation (NSF) Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program. Additionally, we acknowledge the support of Michigan Technological University through the Graduate Student Government (GSG) Finishing Fellowship 2025.

References

[1] International Labor Office (ILO), “Promoting Safe ang Healthy Jobs: The ILO Programme on Safety, Helath and The Environment (Safework),” *World Work*, pp. 1–11, 2008, [Online]. Available: https://www.ilo.org/global/publications/world-of-work-magazine/articles/WCMS_099050/lang--en/index.htm

[2] U.S. BUREAU OF LABOR STATISTICS, “Number and rate of nonfatal work injuries and illnesses in private industries.” [Online]. Available: https://www.bls.gov/charts/injuries-and-illnesses/number-and-rate-of-nonfatal-work-injuries-and-illnesses-by-industry.htm

[3] National Institute for Occupational Safety and Health (NIOSH), “Number and rate of occupational mining fatalities by year, 2000 - 2022.” [Online]. Available: https://wwwn.cdc.gov/NIOSH-Mining/MMWC/Fatality/NumberAndRate?StartYear=2000&EndYear=2022&SelectedOperatorType=&SelectedMineType=

[4] National Institute for Occupational Safety and Health (NIOSH), “Number and rate of nonfatal lost-time injuries, 2000 - 2022.” [Online]. Available: https://wwwn.cdc.gov/NIOSH-Mining/MMWC/Injuries/NumberAndRate?StartYear=2000&EndYear=2022&SelectedOperatorType=&SelectedMineType=

[5] A. Yedla, F. D. Kakhki, and A. Jannesari, “Predictive modeling for occupational safety outcomes and days away from work analysis in mining operations,” *Int. J. Environ. Res. Public Health*, vol. 17, no. 19, pp. 1–17, 2020, doi: 10.3390/ijerph17197054.

[6] R. Ganguli, P. Miller, and R. Pothina, “Effectiveness of natural language processing based machine learning in analyzing incident narratives at a mine,” *Minerals*, vol. 11, no. 7, pp. 1–13, 2021, doi: 10.3390/min11070776.

[7] R. Pothina and R. Ganguli, “The Importance of Specific Phrases in Automatically Classifying Mine Accident Narratives Using Natural Language Processing,” *Knowledge*, vol. 2, no. 3, pp. 365–387, 2022, doi: 10.3390/knowledge2030021.

[8] B. Zhong, X. Pan, P. E. D. Love, L. Ding, and W. Fang, “Deep learning and network analysis: Classifying and visualizing accident narratives in construction,” *Autom. Constr.*, vol. 113, May 2020, doi: 10.1016/j.autcon.2020.103089.

[9] B. Zhong, X. Pan, P. E. D. Love, J. Sun, and C. Tao, “Hazard analysis: A deep learning and text mining framework for accident prevention,” *Adv. Eng. Informatics*, vol. 46, Oct. 2020, doi: 10.1016/j.aei.2020.101152.

[10] N. XU, L. MA, Q. Liu, L. WANG, and Y. Deng, “An improved text mining approach to extract safety risk factors from construction accident reports,” *Saf. Sci.*, vol. 138, Jun. 2021, doi: 10.1016/j.ssci.2021.105216.

[11] P. Srinivasan, V. Nagarajan, and S. Mahadevan, “Mining and classifying aviation accident reports,” *AIAA Aviat. 2019 Forum*, no. June, 2019, doi: 10.2514/6.2019-2938.

[12] R. L. Rose, T. G. Puranik, and D. N. Mavris, “Natural language processing based method for clustering and analysis of aviation safety narratives,” *Aerospace*, vol. 7, no. 10, pp. 1–22, Oct. 2020, doi: 10.3390/aerospace7100143.

[13] T. Williams, J. Betak, and B. Findley, “Text Mining Analysis Of Railroad Accident Investigation Reports,” 2016. [Online]. Available: http://www.tsb.gc.ca/eng/lois-acts/evenements-occurrences.asp.

[14] M. Figueres-Esteban, P. Hughes, and C. van Gulijk, “Visual analytics for text-based railway incident reports,” *Saf. Sci.*, vol. 89, pp. 72–76, Nov. 2016, doi: 10.1016/j.ssci.2016.05.009.

[15] S. Sarkar, S. Vinay, R. Raj, J. Maiti, and P. Mitra, “Application of optimized machine learning techniques for prediction of occupational accidents,” *Comput. Oper. Res.*, vol. 106, pp. 210–224, 2019, doi: 10.1016/j.cor.2018.02.021.

[16] I. Lauriola, A. Lavelli, and F. Aiolli, “An introduction to Deep Learning in Natural Language Processing: Models, techniques, and tools,” *Neurocomputing*, vol. 470, pp. 443–456, 2022, doi: https://doi.org/10.1016/j.neucom.2021.05.103.

[17] M. M. Abedi and E. Sacchi, “A machine learning tool for collecting and analyzing subjective road safety data from Twitter,” *Expert Syst. Appl.*, vol. 240, p. 122582, 2024, doi: https://doi.org/10.1016/j.eswa.2023.122582.

[18] Q. Do, T. Le, and C. Le, “Action Sequencing in Construction Accident Reports using Probabilistic Language Model,” in *Proceedings of the 39th International Symposium on Automation and Robotics in Construction*, T. Linner, B. de Soto, R. Hu, et al., Eds., Bogot, Colombia: International Association for Automation and Robotics in Construction (IAARC), Jul. 2022, pp. 653–660. doi: 10.22260/ISARC2022/0091.

[19] S. Li, M. You, D. Li, and J. Liu, “Identifying coal mine safety production risk factors by employing text mining and Bayesian network techniques,” *Process Saf. Environ. Prot.*, vol. 162, pp. 1067–1081, 2022, doi: https://doi.org/10.1016/j.psep.2022.04.054.

[20] R. L. Rose, T. G. Puranik, D. N. Mavris, and A. H. Rao, “Application of structural topic modeling to aviation safety data,” *Reliab. Eng. Syst. Saf.*, vol. 224, p. 108522, 2022, doi: https://doi.org/10.1016/j.ress.2022.108522.

[21] J. Li and C. Wu, “Deep Learning and Text Mining: Classifying and Extracting Key Information from Construction Accident Narratives,” *Appl. Sci.*, vol. 13, no. 19, 2023, doi: 10.3390/app131910599.

[22] D. M. Goldberg, “Characterizing accident narratives with word embeddings: Improving accuracy, richness, and generalizability,” *J. Safety Res.*, vol. 80, pp. 441–455, 2022, doi: https://doi.org/10.1016/j.jsr.2021.12.024.

[23] J. Pennington, R. Socher, and C. Manning, “GloVe: Global Vectors for Word Representation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, A. Moschitti, B. Pang, and W. Daelemans, Eds., Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. doi: 10.3115/v1/D14-1162.

[24] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” in *International Conference on Learning Representations*, 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:5959482

[25] H. Chen, L. Wu, J. Chen, W. Lu, and J. Ding, “A comparative study of automated legal text classification using random forests and deep learning,” *Inf. Process. Manag.*, vol. 59, no. 2, p. 102798, 2022, doi: https://doi.org/10.1016/j.ipm.2021.102798.

[26] K. N. Singh, S. D. Devi, H. M. Devi, and A. K. Mahanta, “A novel approach for dimension reduction using word embedding: An enhanced text classification approach,” *Int. J. Inf. Manag. Data Insights*, vol. 2, no. 1, p. 100061, 2022, doi: https://doi.org/10.1016/j.jjimei.2022.100061.

[27] M. Heidarysafa, K. Kowsari, L. Barnes, and D. Brown, “Analysis of Railway Accidents’ Narratives Using Deep Learning,” in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2018, pp. 1446–1453. doi: 10.1109/ICMLA.2018.00235.

[28] A. Onan, “Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks,” *Concurr. Comput. Pract. Exp.*, vol. 33, no. 23, p. e5909, 2021, doi: https://doi.org/10.1002/cpe.5909.

[29] F. Zhang, “A hybrid structured deep neural network with Word2Vec for construction accident causes classification,” *Int. J. Constr. Manag.*, vol. 22, no. 6, pp. 1120–1140,

Apr. 2022, doi: 10.1080/15623599.2019.1683692.

[30] X. Luo, X. Li, X. Song, and Q. Liu, “Convolutional Neural Network Algorithm–Based Novel Automatic Text Classification Framework for Construction Accident Reports,” *J. Constr. Eng. Manag.*, vol. 149, no. 12, p. 4023128, 2023, doi: 10.1061/JCEMD4.COENG-13523.

[31] A. K. Gupta, C. G. V. Sai Pardheev, S. Choudhuri, S. Das, A. Garg, and J. Maiti, “A novel classification approach based on context connotative network (CCNet): A case of construction site accidents,” *Expert Syst. Appl.*, vol. 202, p. 117281, 2022, doi: https://doi.org/10.1016/j.eswa.2022.117281.

[32] A. Sherstinsky, “Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network,” *Phys. D Nonlinear Phenom.*, vol. 404, p. 132306, 2020, doi: https://doi.org/10.1016/j.physd.2019.132306.

[33] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling,” *ArXiv*, vol. abs/1412.3, 2014, [Online]. Available: https://api.semanticscholar.org/CorpusID:5201925

[34] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *IEEE Trans. Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994, doi: 10.1109/72.279181.

[35] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” in *International conference on machine learning*, 2013, pp. 1310–1318.

[36] D. Soydaner, “Attention mechanism in neural networks: where it comes and where it goes,” *Neural Comput. Appl.*, vol. 34, no. 16, pp. 13371–13385, 2022, doi: 10.1007/s00521-022-07366-3.

[37] A. Vaswani *et al.*, “Attention is all you need,” *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem, no. Nips, pp. 5999–6009, 2017.

[38] L. Floridi and M. Chiriatti, “GPT-3: Its Nature, Scope, Limits, and Consequences,” *Minds Mach.*, vol. 30, no. 4, pp. 681–694, 2020, doi: 10.1007/s11023-020-09548-1.

[39] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” *NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.*, vol. 1, no. Mlm, pp. 4171–4186, 2019.

[40] Q. Liu, M. J. Kusner, and P. Blunsom, “A survey on contextual embeddings,” *arXiv Prepr. arXiv2003.07278*, 2020.

[41] C. Wei, Y.-C. Wang, B. Wang, and C.-C. J. Kuo, “An overview on language models: Recent developments and outlook,” *arXiv Prepr. arXiv2303.05759*, 2023.

[42] M. O. Topal, A. Bas, and I. van Heerden, “Exploring Transformers in Natural Language Generation: GPT, BERT, and XLNet,” *ArXiv*, vol. abs/2102.0, 2021, [Online]. Available:

https://api.semanticscholar.org/CorpusID:231933669

[43] G. Z. Nabiilah, S. Y. Prasetyo, Z. N. Izdihar, and A. S. Girsang, "BERT base model for toxic comment analysis on Indonesian social media," *Procedia Comput. Sci.*, vol. 216, pp. 714–721, 2023, doi: https://doi.org/10.1016/j.procs.2022.12.188.

[44] S. Srivastava, B. Paul, and D. Gupta, "Study of Word Embeddings for Enhanced Cyber Security Named Entity Recognition," *Procedia Comput. Sci.*, vol. 218, pp. 449–460, 2023, doi: https://doi.org/10.1016/j.procs.2023.01.027.

[45] A. Adhikari, A. Ram, R. Tang, and J. J. Lin, "DocBERT: BERT for Document Classification," *ArXiv*, vol. abs/1904.0, 2019, [Online]. Available: https://api.semanticscholar.org/CorpusID:118697759

[46] Y. Chen and F. Zulkernine, "BIRD-QA: A BERT-based Information Retrieval Approach to Domain Specific Question Answering," in *2021 IEEE International Conference on Big Data (Big Data)*, 2021, pp. 3503–3510. doi: 10.1109/BigData52589.2021.9671523.

[47] Y. Liu and M. Lapata, "Text Summarization with Pretrained Encoders," *ArXiv*, vol. abs/1908.0, 2019, [Online]. Available: https://api.semanticscholar.org/CorpusID:201304248

[48] Z. Wang, P. Ng, X. Ma, R. Nallapati, and B. Xiang, "Multi-passage BERT: A Globally Normalized BERT Model for Open-domain Question Answering," 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)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds., Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 5878–5882. doi: 10.18653/v1/D19-1599.

[49] H. A. Mohamed Hassan, E. Marengo, and W. Nutt, "A BERT-Based Model for Question Answering on Construction Incident Reports," in *Natural Language Processing and Information Systems*, P. Rosso, V. Basile, R. Martínez, E. Métais, and F. Meziane, Eds., Cham: Springer International Publishing, 2022, pp. 215–223.

[50] S. Yuan and Q. Wang, "Imbalanced Traffic Accident Text Classification Based on Bert-RCNN," *J. Phys. Conf. Ser.*, vol. 2170, no. 1, p. 12003, Feb. 2022, doi: 10.1088/1742-6596/2170/1/012003.

[51] A. H. Oliaee, S. Das, J. Liu, and M. A. Rahman, "Using Bidirectional Encoder Representations from Transformers (BERT) to classify traffic crash severity types," *Nat. Lang. Process. J.*, vol. 3, p. 100007, 2023, doi: https://doi.org/10.1016/j.nlp.2023.100007.

[52] V. Linardos, M. Drakaki, and P. Tzionas, "A transformers-based approach on industrial disaster consequence identification from accident narratives," *Procedia Comput. Sci.*, vol. 217, pp. 1446–1451, 2023, doi: https://doi.org/10.1016/j.procs.2022.12.343.

[53] Y. Q. H. H. Bing Song Xiaoping Ma and Z. Zhang, “Railroad accident causal analysis with unstructured narratives using bidirectional encoder representations for transformers,” *J. Transp. Saf. \& Secur.*, vol. 15, no. 7, pp. 717–736, 2023, doi: 10.1080/19439962.2022.2128956.

[54] P. M. S. Ramos, J. B. Macedo, C. B. S. Maior, M. C. Moura, and I. D. Lins, “Combining BERT with numerical variables to classify injury leave based on accident description,” *Proc. Inst. Mech. Eng. Part O J. Risk Reliab.*, p. 1748006X221140194, Dec. 2022, doi: 10.1177/1748006X221140194.

[55] H. Dai, M. Zhu, G. Yuan, Y. Niu, H. Shi, and B. Chen, “Entity Recognition for Chinese Hazardous Chemical Accident Data Based on Rules and a Pre-Trained Model,” *Appl. Sci.*, vol. 13, no. 1, 2023, doi: 10.3390/app13010375.

[56] X. Luo *et al.*, “Extraction and analysis of risk factors from Chinese chemical accident reports,” *Chinese J. Chem. Eng.*, vol. 61, pp. 68–81, 2023, doi: https://doi.org/10.1016/j.cjche.2023.02.026.

[57] J. Lee *et al.*, “BioBERT: a pre-trained biomedical language representation model for biomedical text mining.,” *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, Feb. 2020, doi: 10.1093/bioinformatics/btz682.

[58] I. Beltagy, K. Lo, and A. Cohan, “SciBERT: A Pretrained Language Model for Scientific Text,” in *Conference on Empirical Methods in Natural Language Processing*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:202558505

[59] X. Yang *et al.*, “A large language model for electronic health records,” *npj Digit. Med.*, vol. 5, no. 1, p. 194, 2022, doi: 10.1038/s41746-022-00742-2.

[60] S. Zhou, N. Wang, L. Wang, H. Liu, and R. Zhang, “CancerBERT: a cancer domain-specific language model for extracting breast cancer phenotypes from electronic health records,” *J. Am. Med. Informatics Assoc.*, vol. 29, no. 7, pp. 1208–1216, 2022, doi: 10.1093/jamia/ocac040.

[61] Y. H. Gu, X. Piao, H. Yin, D. Jin, R. Zheng, and S. J. Yoo, “Domain-Specific Language Model Pre-Training for Korean Tax Law Classification,” *IEEE Access*, vol. 10, pp. 46342–46353, 2022, doi: 10.1109/ACCESS.2022.3164098.

[62] D. Araci, “Finbert: Financial sentiment analysis with pre-trained language models,” *arXiv Prepr. arXiv1908.10063*, 2019.

[63] Y. Yang, M. C. S. Uy, and A. Huang, “FinBERT: A Pretrained Language Model for Financial Communications,” *ArXiv*, vol. abs/2006.0, 2020, [Online]. Available: https://api.semanticscholar.org/CorpusID:219687757

[64] Y. Zhong and S. D. Goodfellow, “Domain-specific language models pre-trained on construction management systems corpora,” *Autom. Constr.*, vol. 160, p. 105316, 2024, doi: https://doi.org/10.1016/j.autcon.2024.105316.

[65] M. Bayer, P. D. Kuehn, R. Shanehsaz, and C. A. Reuter, “CySecBERT: A Domain-Adapted Language Model for the Cybersecurity Domain,” *ArXiv*, vol. abs/2212.0, 2022, [Online]. Available: https://api.semanticscholar.org/CorpusID:254275230

[66] T. Caselli, V. Basile, J. Mitrović, and M. Granitzer, “HateBERT: Retraining BERT for Abusive Language Detection in English,” *ArXiv*, vol. abs/2010.1, 2020, [Online]. Available: https://api.semanticscholar.org/CorpusID:225062242

[67] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, “LEGAL-BERT: “Preparing the Muppets for Court’”,” *ArXiv*, vol. abs/2010.0, 2020, [Online]. Available: https://api.semanticscholar.org/CorpusID:222141043

[68] Z. Zheng, X.-Z. Lu, K.-Y. Chen, Y.-C. Zhou, and J.-R. Lin, “Pretrained domain-specific language model for natural language processing tasks in the AEC domain,” *Comput. Ind.*, vol. 142, p. 103733, 2022, doi: https://doi.org/10.1016/j.compind.2022.103733.

[69] C. J. M. Lawley *et al.*, “Geoscience language models and their intrinsic evaluation,” *Appl. Comput. Geosci.*, vol. 14, p. 100084, 2022, doi: https://doi.org/10.1016/j.acags.2022.100084.

[70] C. Chandra *et al.*, “Aviation-BERT: A Preliminary Aviation-Specific Natural Language Model,” in *AIAA AVIATION 2023 Forum*, 2023. doi: 10.2514/6.2023-3436.

[71] S. Kierszbaum, T. Klein, and L. Lapasset, “ASRS-CMFS vs. RoBERTa: Comparing Two Pre-Trained Language Models to Predict Anomalies in Aviation Occurrence Reports with a Low Volume of In-Domain Data Available,” *Aerospace*, vol. 9, no. 10, 2022, doi: 10.3390/aerospace9100591.

[72] R. Pothina and R. Ganguli, “Contextual Representation in NLP to Improve Success in Accident Classification of Mine Safety Narratives,” *Minerals*, vol. 13, no. 6, 2023, doi: 10.3390/min13060770.

[73] B. Peng, E. Chersoni, Y. Y. Hsu, and C.-R. Huang, “Is Domain Adaptation Worth Your Investment? Comparing BERT and FinBERT on Financial Tasks,” in *Proceedings of the Third Workshop on Economics and Natural Language Processing*, U. Hahn, V. Hoste, and A. Stent, Eds., Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 37–44. doi: 10.18653/v1/2021.econlp-1.5.

[74] S. Gururangan *et al.*, “Don’t Stop Pretraining: Adapt Language Models to Domains and Tasks,” *ArXiv*, vol. abs/2004.1, 2020, [Online]. Available: https://api.semanticscholar.org/CorpusID:216080466

[75] H.-C. Shin *et al.*, “Bio-Megatron: Larger Biomedical Domain Language Model,” *ArXiv*, vol. abs/2010.0, 2020, [Online]. Available: https://api.semanticscholar.org/CorpusID:222310618

[76] M. Suzuki, H. Sakaji, M. Hirano, and K. Izumi, “Constructing and analyzing domain-specific language model for financial text mining,” *Inf. Process. Manag.*, vol. 60, no. 2, p. 103194, 2023, doi: https://doi.org/10.1016/j.ipm.2022.103194.

[77] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, “ALBERT: A Lite BERT for Self-supervised Learning of Language Representations,” *ArXiv*, vol. abs/1909.1, 2019, [Online]. Available: https://api.semanticscholar.org/CorpusID:202888986

[78] H. Choi, J. Kim, S. Joe, and Y. Gwon, “Evaluation of BERT and ALBERT Sentence Embedding Performance on Downstream NLP Tasks,” in *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, pp. 5482–5487. doi: 10.1109/ICPR48806.2021.9412102.

[79] C. Özkurt, “Comparative Analysis of State-of-the-Art Q&A Models: BERT, RoBERTa, DistilBERT, and ALBERT on SQuAD v2 Dataset,” *Chaos and Fractals*, vol. 1, no. 1 SE-Articles, pp. 19–30, Jul. 2024, doi: 10.69882/adba.chf.2024073.

[80] A. Areshey and H. Mathkour, “Exploring transformer models for sentiment classification: A comparison of BERT, RoBERTa, ALBERT, DistilBERT, and XLNet,” *Expert Syst.*, vol. 41, no. 11, 2024, doi: https://doi.org/10.1111/exsy.13701.

[81] K. raj Kanakarajan, B. Kundumani, and M. Sankarasubbu, “BioELECTRA:Pretrained Biomedical text Encoder using Discriminators,” in *Proceedings of the 20th Workshop on Biomedical Language Processing*, D. Demner-Fushman, K. B. Cohen, S. Ananiadou, and J. Tsujii, Eds., Online: Association for Computational Linguistics, Jun. 2021, pp. 143–154. doi: 10.18653/v1/2021.bionlp-1.16.

[82] S. Alrowili and V. Shanker, “BioM-Transformers: Building Large Biomedical Language Models with BERT, ALBERT and ELECTRA,” in *Proceedings of the 20th Workshop on Biomedical Language Processing*, D. Demner-Fushman, K. B. Cohen, S. Ananiadou, and J. Tsujii, Eds., Online: Association for Computational Linguistics, Jun. 2021, pp. 221–227. doi: 10.18653/v1/2021.bionlp-1.24.

[83] S. Ji, T. Zhang, L. Ansari, J. Fu, P. Tiwari, and E. Cambria, “MentalBERT: Publicly Available Pretrained Language Models for Mental Healthcare,” in *International Conference on Language Resources and Evaluation*, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:240288892

[84] U. Naseem, A. G. Dunn, M. Khushi, and J. Kim, “Benchmarking for biomedical natural language processing tasks with a domain specific ALBERT,” *BMC Bioinformatics*, vol. 23, no. 1, p. 144, 2022, doi: 10.1186/s12859-022-04688-w.

[85] H. El Boukkouri, O. Ferret, T. Lavergne, and P. Zweigenbaum, “Re-train or Train from Scratch? Comparing Pre-training Strategies of BERT in the Medical Domain,” *2022 Lang. Resour. Eval. Conf. Lr. 2022*, no. June, pp. 2626–2633, 2022.

[86] Y. Gu *et al.*, “Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing,” *ACM Trans. Comput. Healthc.*, vol. 3, no. 1, Oct. 2021, doi: 10.1145/3458754.

[87] E. Kazan and M. A. Usmen, “Worker safety and injury severity analysis of

earthmoving equipment accidents," *J. Safety Res.*, vol. 65, pp. 73–81, 2018, doi: https://doi.org/10.1016/j.jsr.2018.02.008.

[88] S. Chi, H. Sangwon, K. Dae Young, and Y. and Shin, "Accident risk identification and its impact analyses for strategic construction safety management," *J. Civ. Eng. Manag.*, vol. 21, no. 4, pp. 524–538, May 2015, doi: 10.3846/13923730.2014.890662.

[89] L. G. Cuenca, E. Puertas, N. Aliane, and J. F. Andres, "Traffic Accidents Classification and Injury Severity Prediction," in *2018 3rd IEEE International Conference on Intelligent Transportation Engineering (ICITE)*, 2018, pp. 52–57. doi: 10.1109/ICITE.2018.8492545.

[90] S. Shrestha, S. A. Morshed, N. Pradhananga, and X. Lv, "Leveraging accident investigation reports as leading indicators of construction safety using text classification," in *Construction Research Congress 2020*, American Society of Civil Engineers Reston, VA, 2020, pp. 490–498.

[91] J.-H. Song, S.-H. Shin, S.-Y. Kang, J.-H. Won, and K.-H. Yoo, "Occurrence Type Classification for Establishing Prevention Plans Based on Industrial Accident Cases Using the KoBERT Model," *Applied Sciences*, vol. 14, no. 20. 2024. doi: 10.3390/app14209450.

[92] M. A. Ansari, G. Ishigaki, and W. B. Andreopoulos, "Fine-Tuning Large Language Models for Environmental Health and Safety Applications," in *2024 Conference on AI, Science, Engineering, and Technology (AIxSET)*, 2024, pp. 45–52. doi: 10.1109/AIxSET62544.2024.00012.

[93] H. Touvron *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv Prepr. arXiv2307.09288*, 2023.

[94] A. Grattafiori *et al.*, "The llama 3 herd of models," *arXiv Prepr. arXiv2407.21783*, 2024.

[95] C. Liu *et al.*, "More Than Catastrophic Forgetting: Integrating General Capabilities For Domain-Specific LLMs," *arXiv Prepr. arXiv2405.17830*, 2024.

[96] K. Arumae and P. Bhatia, "CALM: Continuous Adaptive Learning for Language Modeling," *arXiv Prepr. arXiv2004.03794*, 2020.

[97] S. Rongali, A. N. Jagannatha, B. P. S. Rawat, and H. Yu, "Improved Pretraining for Domain-specific Contextual Embedding Models," *ArXiv*, vol. abs/2004.0, 2020, [Online]. Available: https://api.semanticscholar.org/CorpusID:214802823

[98] A. K. Ohm and K. K. Singh, "Study of Tokenization Strategies for the Santhali Language," *SN Comput. Sci.*, vol. 5, no. 7, p. 807, 2024, doi: 10.1007/s42979-024-03083-x.

[99] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv Prepr.*

arXiv1810.04805, 2018.

[100] A. Asvarov and A. Grabovoy, “The Impact of Multilinguality and Tokenization on Statistical Machine Translation,” in *2024 35th Conference of Open Innovations Association (FRUCT)*, 2024, pp. 149–157. doi: 10.23919/FRUCT61870.2024.10516416.

[101] Y. Zhai *et al.*, “ ByteTransformer: A High-Performance Transformer Boosted for Variable-Length Inputs ,” in *2023 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, Los Alamitos, CA, USA: IEEE Computer Society, May 2023, pp. 344–355. doi: 10.1109/IPDPS54959.2023.00042.

[102] M. M. Krell, M. Kosec, S. P. Perez, and A. Fitzgibbon, “Efficient Sequence Packing Without Cross-Contamination: Accelerating Large Language Models Without Impacting Performance,” *arXiv Prepr. arXiv2107.02027*, 2021.

[103] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V Le, “XLNET: Generalized Autoregressive Pretraining for Language Understanding,” *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

[104] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, “Spanbert: Improving Pre-training by Representing and Predicting Spans,” *Trans. Assoc. Comput. Linguist.*, vol. 8, pp. 64–77, 2020.

[105] Y. Liu *et al.*, “Roberta: A Robustly Optimized Bert Pretraining Approach,” *arXiv Prepr. arXiv1907.11692*, 2019.

[106] V. Sachidananda, J. S. Kessler, and Y.-A. Lai, “Efficient Domain Adaptation of Language Models via Adaptive Tokenization,” *arXiv Prepr. arXiv2109.07460*, 2021.

[107] P. Micikevicius *et al.*, “Mixed Precision Training,” *arXiv Prepr. arXiv1710.03740*, 2017.

[108] J. Salazar, D. Liang, T. Q. Nguyen, and K. Kirchhoff, “Masked Language Model Scoring,” *arXiv Prepr. arXiv1910.14659*, 2019.

[109] S. Chatterjee, K. Poorva, K. Rennie, M. Hugh, and A. and Majdara, “Risk factors identification and injury severity classification in Alaska’s mining industry using statistical and machine learning approaches,” *Int. J. Mining, Reclam. Environ.*, pp. 1–18, doi: 10.1080/17480930.2025.2459238.

[110] L. Prechelt, “Automatic early stopping using cross validation: quantifying the criteria.,” *Neural networks Off. J. Int. Neural Netw. Soc.*, vol. 11, no. 4, pp. 761–767, Jun. 1998, doi: 10.1016/s0893-6080(98)00010-0.

[111] S. Ruder, “An overview of multi-task learning in deep neural networks,” *arXiv Prepr. arXiv1706.05098*, 2017.

[112] A. Kendall, Y. Gal, and R. Cipolla, “Multi-task learning using uncertainty to weigh losses for scene geometry and semantics,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7482–7491.

[113] Y. Zhang and Q. Yang, “A survey on multi-task learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 12, pp. 5586–5609, 2021.

[114] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” *J. Mach. Learn. Res.*, vol. 13, no. null, pp. 281–305, Feb. 2012.