ManufactuBERT: Efficient Continual Pretraining for Manufacturing

Robin Armingaud, Romaric Besançon
Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France
{firstname.lastname}@cea.fr

Abstract

While large general-purpose Transformer-based encoders excel at general language understanding, their performance diminishes in specialized domains like manufacturing due to a lack of exposure to domain-specific terminology and semantics. In this paper, we address this gap by introducing ManufactuBERT, a RoBERTa model continually pretrained on a large-scale corpus curated for the manufacturing domain. We present a comprehensive data processing pipeline to create this corpus from web data, involving an initial domain-specific filtering step followed by a multi-stage deduplication process that removes redundancies. Our experiments show that ManufactuBERT establishes a new state-of-the-art on a range of manufacturing-related NLP tasks, outperforming strong specialized baselines. More importantly, we demonstrate that training on our carefully deduplicated corpus significantly accelerates convergence, leading to a 33% reduction in training time and computational cost compared to training on the non-deduplicated dataset. The proposed pipeline offers a reproducible example for developing high-performing encoders in other specialized domains. We will release our model and curated corpus at https://huggingface.co/cea-list-ia.

Keywords: Manufacturing, NLP, Domain Adaptation

1. Introduction

The digitalization of the manufacturing sector has led to an explosion of textual data, making Natural Language Processing (NLP) methods especially efficient for tasks such as automated anomaly detection, technical report completion and knowledge extraction (May et al., 2022; Bernabei et al., 2022; Li et al., 2024). Transformer-based encoders, from BERT(Devlin et al., 2019) to its successors RoBERTa (Liu et al., 2019), DeBERTa (He et al., 2021a) and more recently NeoBERT (Breton et al., 2025) and ModernBERT (Warner et al., 2024), have achieved near-human-level performance on a wide range of language understanding benchmarks, including Named Entity Recognition (NER) and Relation Extraction (RE). However, the effectiveness of these general-purpose models is often limited when applied to specialized domains. Industrial and manufacturing texts exhibit unique linguistic properties, including specialized terminology, a high frequency of acronyms and context-dependent meanings that differ from common usage. The standard approach to bridge this gap is domain adaptation (Gururangan et al., 2020), which involves continuing the pretraining of a model on a large, domain-specific corpus. While effective, this process is computationally expensive, requiring significant resources and contributing to a larger carbon footprint.

This work addresses the challenge of efficiently adapting language models to the manufacturing domain. We introduce a pipeline for creating a high-quality, domain-specific corpus and a new pretrained language model, ManufactuBERT. We propose the following contributions :

- We introduce ManufactuBERT, a new Pretrained Language Model (PLM) based on RoBERTa, pretrained on a large-scale, curated corpus of manufacturing-related texts.
- We propose an efficient adaptation pipeline for encoder-based language models, designed to reduce the data and energy footprint of domain specialization : we construct a manufacturing-oriented pretraining corpus by filtering the FineWeb dataset (Penedo et al., 2024a) using a fastText-based domain classifier (Joulin et al., 2017), followed by deduplication using SemDeDup (Abbas et al., 2023). This approach yields a compact yet representative dataset that accelerates training convergence and reduces storage requirements.
- We conduct a comprehensive evaluation of ManufactuBERT against strong baselines on

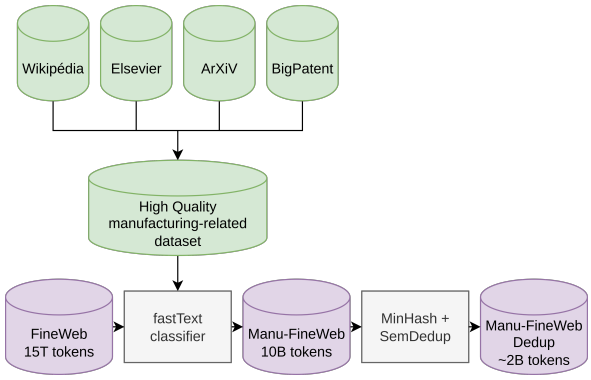

Figure 1: Workflow of the data filtering and deduplication steps used to create the ManufactuBERT pretraining corpus.

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 10545–10555
11–16 May 2026.

a suite of tasks relevant to the manufacturing domain, including the FabNER benchmark (Kumar and Starly, 2022). Our results show that ManufactuBERT establishes a new state-of-the-art on several of these tasks.

2. Related Work

Transformer models Since their introduction, encoder-only Transformer models like BERT (Devlin et al., 2019), have become foundational in NLP. This architecture has inspired a family of successors. For instance, RoBERTa (Liu et al., 2019) revisited BERT's pretraining strategy, showing that removing the next-sentence prediction objective, increasing batch sizes, using dynamic masking and training on more data yields stronger performance. DeBERTa / DeBERTaV3 (He et al., 2021b; He et al., 2021a) improves upon previous encoders by introducing disentangled attention, enhanced position encodings and more efficient hyperparameters.

More recently, models such as ModernBERT and NeoBERT have started to incorporate more training data and architectural concepts from modern Large Language Models (LLMs) into the BERT framework. Despite these advances, a common limitation persists: these models are trained on general-domain corpora, which restricts their applicability in specialized fields without further adaptation to learn domain-specific vocabularies and semantic nuances.

Domain Adaptation One of the standard approaches to adapt a general-purpose PLM to a specialized domain is continued pretraining on in-domain unlabeled text, referred to as domain-adaptive pretraining. This involves continuing the Masked Language Modeling (MLM) pretraining objective. Gururangan et al. (2020) provided strong evidence that domain-adaptive pretraining yields performance gains on downstream tasks.

This principle has led to the development of numerous domain-specific models. Notable examples include BioBERT (Lee et al., 2019) which adapts BERT by further pretraining on large corpora from PubMed abstracts and PMC full texts or FinBERT (Liu et al., 2020) which adapt BERT to financial domain texts. Some models, like SciBERT (Beltagy et al., 2019), are trained from scratch on scientific publications, allowing for the creation of a new and domain-aligned vocabulary.

Closer to our work, MatSciBERT (Gupta et al., 2021) specializes in materials science by continuing the pretraining of SciBERT on a targeted corpus of scientific articles.

While alternative adaptation methods exist, such as vocabulary expansion techniques like AVocaDo (Hong et al., 2021), they have shown limited efficacy with more robust encoders like RoBERTa. Furthermore, Kim et al. (2024) suggests that these methods may offer limited benefits in highly technical domains like materials science , without additional tuning.

Data Selection for Pretraining The high cost of pretraining is linked to the massive scale of the datasets used, which are often derived from web crawls like Common Crawl such as C4 (Raffel et al., 2019) or more recently FineWeb (Abbas et al., 2023) and RefinedWeb (Penedo et al., 2023). While these corpora apply extensive filtering, they still contain significant redundancy. To address this issue, recent work has focused on optimizing data selection to improve training efficiency and model performance. This has led to the development of sophisticated deduplication algorithms that go beyond simple lexical matching such as Minhash (Broder, 1997). Approaches like SemDeDup (Abbas et al., 2023), Density Based Pruning (Abbas et al., 2024) and D4 (Tirumala et al., 2023) use embeddings to identify and remove similar documents and increase data diversity. While these methods have primarily been applied to image models or LLMs, we employ SemDeDup to clean our pretraining corpus, and to the best of our knowledge, we are the first to use this approach in the context of MLM.

3. Methodology

3.1. Domain-specific pretraining dataset construction

While many domain-specific encoders, such as SciBERT, are pretrained on curated and homogeneous corpora like scientific literature, which leads to excellent performance on academic benchmarks built on the same corpora such as SciERC or SciCite (Luan et al., 2018; Cohan et al., 2019), this approach can create a domain mismatch with the linguistic diversity found in real-world industrial applications. To pretrain ManufactuBERT, we instead rely on a specific selection and curation process from a large-scale, web-based diverse corpus. This process, illustrated in Figure 1 is composed of two steps: filtering and deduplication.

3.1.1. Data Filtering

We select as base corpus the recent and high-quality FineWeb dataset (Penedo et al., 2024a), derived from Common Crawl and comprising approximately 15 trillion tokens. We then filter this dataset using a classifier trained to identify documents that are relevant to the manufacturing domain. We use a FastText classifier (Joulin et al.,

Data Source	Selection Criteria	Documents
Elsevier	Abstracts from manufacturing-related journals retrieved via the Elsevier API, based on the SciMago journal rankings for the "Industrial and Manufacturing Engineering" category.	27 943
ArXiv	Abstracts from the cond-mat, physics and eess categories containing keywords such as *manufacturing*, *3D printing*, or *industrial process*, following a similar methodology to Kumar and Starly (2022).	2 042
Wikipedia	Articles from categories including "Manufacturing", "Engineering" and "Industrial processes".	5 907
BigPatent (Sharma et al., 2019)	Patent descriptions containing the keyword "manufacturing".	26 428

Table 1: Training data for the manufacturing-domain classifier used to filter FineWeb.

2017), chosen for its efficiency and effectiveness in tasks such as domain or language identification. The classifier is trained on a curated dataset with positive examples selected from different relevant sources and negative examples randomly selected from FineWeb, with a negative-to-positive ratio of 10:1. The sources and criteria used for the positive examples are summarized in Table 1. Note that this dataset is too small to support effective pretraining and we use it only to train the classifier. To ensure the high quality of our domain-specific corpus, we evaluate the FastText classifier's performance under a 1:10 positive-to-negative class ratio. As summarized in Table 2, the classifier demonstrates strong discriminative capabilities. Notably, it achieves high precision, 0.96, for the manufacturing class.

Class Label	Precision	Recall	F1-Score
Manufacturing	0.96	0.84	0.90
Other	0.98	1.00	0.99

Table 2: Detailed classification performance of the FastText filter.

We filter FineWeb using the trained classifier and increase the prediction threshold to 0.7 to obtain a corpus of approximately 10 billion tokens (around 21 million documents), comparable in scale to the corpora used by Gururangan et al. (2020).

3.1.2. Data Deduplication

Since FineWeb is composed of multiple individually deduplicated Common Crawl snapshots, it still contains a substantial amount of residual duplicates. To further improve the quality and diversity of our pretraining corpus, we apply SemDeDup (Abbas et al., 2023), a deduplication algorithm designed to identify and remove semantically redundant documents. Following the authors' recommendations, we first perform MinHash deduplication with 20 buckets and 20 signatures per bucket to eliminate exact and near-lexical duplicates. The SemDeDup process then consists of three main steps:

1. Computing a semantic vector representation for each document in the corpus. For this task, we employ the all-MiniLM-L6-v2 sentence encoder, a computationally efficient and widely-used model for semantic similarity tasks. To handle documents that exceed the model's maximum input length, we segment each document into chunks of 512 tokens. Each chunk is independently encoded using the Sentence-Transformers library[1], and the resulting chunk embeddings are then averaged to produce a single vector representing the entire document.

2. Partitioning the embeddings into n clusters using K-means

3. Within each cluster, pruning any document that is closer to another document than a specified distance threshold τ.

For our final corpus, we set the number of clusters $n = 1000$ and the distance threshold $\tau = 0.15$. This combined MinHash and SemDeDup process effectively removes approximately 80% of the documents from our initial filtered manufacturing dataset. We implement this pipeline using the Datatrove library (Penedo et al., 2024b), originally released with the FineWeb dataset, along with a custom implementation of SemDeDup. The resulting corpus of approximately 4.5 million documents is highly aligned with the target domain while maintaining strong general-domain performance, as demonstrated in Section 4. Moreover, the proposed pipeline is easily reproducible for other domains, provided that a domain classifier can be trained.

[1] https://www.sbert.net/

3.2. Masked Language Modeling Pretraining

We perform continued pretraining, initializing our model with the publicly available RoBERTa-base checkpoint. The training follows the standard Masked Language Modeling objective, omitting the Next Sentence Prediction task, which is consistent with the methodology of RoBERTa.

We adopt the hyperparameters from Gururangan et al. (2020): a batch size of 16 with 16 gradient accumulation steps (for an effective batch size of 256 per GPU and 2048 in total), a masking probability of 0.15, a weight decay of 0.1 and a maximum learning rate of 5×10^{-4} with a linear scheduler and 6% warmup steps. The model is checkpointed every 500 steps. We make one modification: we extend the training schedule from 12,500 to 17,500 steps, as our preliminary experiments showed that the model had not yet converged at 12,500 steps. This pretraining phase was executed on a node of 8 NVIDIA V100-32GB GPUs and required approximately 51 hours of computation. We pretrain two models: ManufactuBERT, using the non-deduplicated dataset and ManufactuBERT$_D$, using the deduplicated corpus. The latter converges faster and achieves higher performance.

4. Experiments

4.1. Datasets

To evaluate our model, we select a range of datasets related to the manufacturing domain. Due to the limited availability of annotated resources in this area, we also include datasets from adjacent technical domains to assess our model's generalization across diverse tasks. Our selection criteria include data accessibility, task diversity and domain proximity. The chosen datasets span three core NLP tasks: Sentence Classification (SC), Named Entity Recognition (NER) and Relation Extraction (RE).

We evaluate on the following benchmarks:

- **FabNER** (Kumar and Starly, 2022) : A manufacturing-domain corpus for NER, consisting of approximately 14k scientific abstracts.
- **Materials Synthesis** (Mysore et al., 2019) : A dataset of 230 abstracts describing materials synthesis procedures. Its labeled graphs enable both NER and RE evaluation.
- **SOFC** (Friedrich et al., 2020) : A corpus of 45 expert-annotated research articles on solid oxide fuel cells (SOFCs). The dataset defines four NER entity types,Material, Experiment, Value and Device and an extended slot-filling version (SOFC-Slot) with 16 fine-grained entity types. It also includes sentence-level annotations for SC, distinguishing SOFC-related and non-related content.
- **MatScholar** (Weston et al., 2019) : A hand-annotated corpus of 800 materials science abstracts for NER.
- **Big Patent** (Sharma et al., 2019) : A large-scale dataset of U.S. patents categorized into nine classes. We randomly sample 1,000 documents per class for training, 200 for validation and 200 for testing, and perform document classification based on the abstracts.
- **ChemdNER** (Krallinger et al., 2015) : A dataset of 10,000 PubMed abstracts annotated for chemical entities. While originating from the biomedical domain, this corpus serves as a benchmark to evaluate our model's ability to identify chemical terms. It is useful for advanced manufacturing sectors, such as pharmaceutical manufacturing.

While our focus is on domain-specific performance, it is also crucial to assess whether the adaptation process has degraded the model's language understanding capabilities in general-domain. To this end, we also evaluate ManufactuBERT on the General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2019). This evaluation serves to quantify any potential loss in general-domain performance and allows for a direct comparison with other specialized models, MatSciBERT and SciBERT. Following standard evaluation practices, we exclude the WNLI task from the benchmark.

4.2. Experimental Setup

For all manufacturing-related downstream tasks, we fine-tune each model for a maximum of 20 epochs, reporting results across 5 different random seeds. For each run, we select the checkpoint that achieves the highest performance on the development set for final evaluation.

For the GLUE benchmark, we fine-tune the models for 10 epochs on each dataset. For all experiments, we report the mean and standard deviation of the results. Our implementation is based on the HuggingFace Transformers library (Wolf et al., 2020). For GLUE, we use the official text classification script provided in the Transformers repository[2].

We use a fixed learning rate of 2×10^{-5}, a batch size of 16, a weight decay of 0.1, the AdamW optimizer (Loshchilov and Hutter, 2019) and a linear

[2] https://github.com/huggingface/transformers/blob/main/examples/pytorch/text-classification/run_glue.py

learning rate scheduler similar to Gururangan et al. (2020) experiments.

All experiments are conducted on a single NVIDIA A100 or H100 GPU.

4.3. Results

4.3.1. Manufacturing-Related Tasks

We conduct a comprehensive evaluation of our models on nine manufacturing-related tasks. The models are benchmarked against a suite of strong baselines:

- General-purpose encoders: RoBERTa (Liu et al., 2019), DeBERTaV3 (He et al., 2021a), NeoBERT (Breton et al., 2025) and ModernBERT (Warner et al., 2024)
- Domain-specific models : SciBERT (Beltagy et al., 2019) and MatSciBERT (Gupta et al., 2021).

To ensure a fair comparison, we use the base version for all models. The detailed results, reported as micro F1 scores, are presented in Table 4 and Table 3.[3]

We group classification-related tasks (Sentence Classification and Relation Extraction) in Table 4 and NER tasks in Table 3. The results demonstrate the effectiveness of our domain adaptation strategy. Our primary model, ManufactuBERT$_D$, achieves the highest average F1 score across both groups of tasks, outperforming the strongest domain-specific baseline, MatSciBERT, and the original RoBERTa model. This confirms that our pretraining corpus effectively captures knowledge relevant to the manufacturing domain. ManufactuBERT$_D$ establishes new state-of-the-art performance on four of the nine evaluated tasks.

Moreover, the model trained on the semantically deduplicated corpus, ManufactuBERT$_D$, consistently outperforms its counterpart trained on the non-deduplicated data ManufactuBERT. This result underscores the value of using SemDeDup for MLM-based domain adaptation, as it leads to better downstream performance even with a smaller pretraining dataset.

To verify the statistical significance of these improvements, we perform a pairwise model comparison using the Almost Stochastic Order (ASO) test (Del Barrio et al., 2018; Dror et al., 2019), following the implementation by Ulmer et al. (2022). With a confidence level of α=0.05 (adjusted with a Bonferroni correction) and τ=0.5, the results confirm the superiority of ManufactuBERT$_D$. It is stochastically dominant over its non-deduplicated version on 5 tasks and over the next best model, MatSciBERT, on 5 tasks. Its advantage is even more pronounced when compared to general-purpose models, where it dominates RoBERTa and NeoBERT on all nine tasks and demonstrates clear superiority over the remaining baselines on the majority of tasks.[4]

4.3.2. GLUE

To quantify the impact of domain adaptation on general language understanding capabilities or *catastrophic forgetting*, we evaluate our models on the GLUE benchmark. The primary goal is to measure the degree of performance degradation relative to our starting checkpoint, RoBERTa, and to compare this degradation against that of other specialized models, SciBERT and MatSciBERT.

The results, summarized in Table 5, indicate that while our models expectedly exhibit a performance drop compared to the general-domain RoBERTa baseline, they retain their foundational language abilities far more effectively than other specialized models. Both ManufactuBERT and ManufactuBERT$_D$ consistently and significantly outperform SciBERT and MatSciBERT on every GLUE task. On average, our models score over 4.5 points higher than MatSciBERT and 3.6 points higher than SciBERT on the same experimental setup.

This result shows that our adaptation recipe is effective. By continuing pretraining on a broad but domain-filtered web corpus, we preserve general language knowledge while adding manufacturing-specific expertise. In contrast, models like SciBERT, trained from scratch on academic text, tend to overfit to their domain and lose general understanding. Our approach achieves a better balance between domain knowledge and general language ability.

5. Analysis

5.1. Data Deduplication Cost

To quantify the impact of our data curation on training efficiency, we analyzed the convergence speed of ManufactuBERT (trained on the filtered

[3]ManuBERT (Kumar et al., 2023), a non-peer-reviewed model pretrained for the manufacturing domain, was excluded from our main benchmark as its reported results could not be reproduced using the publicly available checkpoint available on Huggingface : https://huggingface.co/akumar33/ManuBERT .

[4]Recent models NeoBERT and ModernBERT exhibit unexpectedly low performance on NER tasks. MMBERT (Marone et al., 2025), the multilingual extension of ModernBERT, is the only model in this family to be evaluated on NER; the authors report inferior results compared to older models and attribute this deficiency to a pretokenizer limitation and specifically the omission of prefix spaces.

Model	Materials Synthesis	FabNER	SOFC		MatScholar	ChemdNER	
	NER	NER	NER	NER SLOT	NER	NER	Avg.
ModernBERT	$69.60_{\pm 1.71}$	$78.76_{\pm 0.68}$	$74.62_{\pm 1.43}$	$53.44_{\pm 1.80}$	$80.14_{\pm 0.46}$	$89.12_{\pm 0.25}$	74.28
NeoBERT	$73.00_{\pm 1.53}$	$82.36_{\pm 0.27}$	$74.14_{\pm 0.89}$	$57.90_{\pm 4.08}$	$81.56_{\pm 0.57}$	$90.40_{\pm 0.23}$	76.56
RoBERTa	$73.12_{\pm 0.35}$	$82.48_{\pm 0.33}$	$82.54_{\pm 0.88}$	$69.52_{\pm 1.52}$	$84.04_{\pm 0.18}$	$90.50_{\pm 0.20}$	80.37
SciBERT	$77.72_{\pm 0.41}$	$83.60_{\pm 0.28}$	$80.34_{\pm 0.48}$	$69.10_{\pm 0.68}$	$84.52_{\pm 0.13}$	$91.80_{\pm 0.10}$	81.18
DeBERTaV3	$73.92_{\pm 0.54}$	$\mathbf{84.62}_{\pm 0.16}$	$82.86_{\pm 0.77}$	$70.68_{\pm 0.98}$	$85.04_{\pm 0.49}$	$91.74_{\pm 0.17}$	81.48
MatSciBERT	$\mathbf{76.50}_{\pm 0.87}$	$83.88_{\pm 0.19}$	$82.10_{\pm 0.57}$	$72.60_{\pm 1.34}$	$85.88_{\pm 0.32}$	$\mathbf{92.00}_{\pm 0.19}$	82.16
ManufactuBERT	$75.20_{\pm 0.60}$	$83.88_{\pm 0.18}$	$83.96_{\pm 0.76}$	$73.64_{\pm 0.85}$	$86.06_{\pm 0.32}$	$91.94_{\pm 0.21}$	82.45
ManufactuBERT$_D$	$75.04_{\pm 0.22}$	$84.00_{\pm 0.24}$	$\mathbf{84.40}_{\pm 0.34}$	$\mathbf{73.68}_{\pm 1.05}$	$\mathbf{86.76}_{\pm 0.34}$	$91.92_{\pm 0.16}$	**82.63**

Table 3: **Manufacturing Domain NER**: Micro-averaged F1 scores across six manufacturing-related NER tasks. ManufactuBERT$_D$ denotes the model trained on the deduplicated dataset. Best results per column are shown in **bold**.

Model	Materials Synthesis	SOFC	Big Patent	
	RE	SC	SC	Avg.
ModernBERT	$95.10_{\pm 0.22}$	$94.36_{\pm 0.26}$	$63.12_{\pm 0.24}$	84.19
NeoBERT	$94.36_{\pm 0.37}$	$94.46_{\pm 0.15}$	$64.58_{\pm 0.63}$	84.47
RoBERTa	$94.32_{\pm 0.22}$	$94.32_{\pm 0.15}$	$63.58_{\pm 0.68}$	84.07
SciBERT	$95.34_{\pm 0.17}$	$\mathbf{94.76}_{\pm 0.18}$	$63.98_{\pm 0.46}$	84.69
DeBERTaV3	$\mathbf{95.80}_{\pm 0.20}$	$94.40_{\pm 0.16}$	$64.46_{\pm 0.54}$	84.89
MatSciBERT	$95.48_{\pm 0.26}$	$94.72_{\pm 0.34}$	$63.80_{\pm 0.58}$	84.67
ManufactuBERT	$94.60_{\pm 0.22}$	$94.46_{\pm 0.15}$	$65.50_{\pm 0.35}$	84.85
ManufactuBERT$_D$	$94.62_{\pm 0.11}$	$94.68_{\pm 0.13}$	$\mathbf{65.80}_{\pm 0.31}$	**85.03**

Table 4: **Manufacturing Domain RE and SC**: Micro-averaged F1 scores across three manufacturing-related classification tasks. ManufactuBERT$_D$ denotes the model trained on the deduplicated dataset. Best results per column are shown in **bold**.

Model	CoLA	MNLI	MRPC	SST-2	QNLI	RTE	STS-B	QQP	Avg.
General Domain Model									
RoBERTa	63.6	87.6	90.2	94.8	92.8	78.7	91.2	91.9	86.35
Specialized Models									
MatSciBERT	34.27	81.03	84.40	88.91	88.23	62.58	85.83	90.73	77.00
SciBERT	37.84	81.46	85.78	88.42	89.85	62.21	88.50	90.94	78.13
ManufactuBERT	**50.89**	**85.35**	85.78	**92.66**	**91.49**	68.11	89.17	91.28	**81.84**
ManufactuBERT$_D$	49.52	85.18	**87.25**	91.40	91.13	**68.59**	**89.80**	**91.36**	81.78

Table 5: **GLUE benchmark results**: We report accuracy for MNLI, MRPC, QNLI, QQP, RTE and SST-2, Matthew's correlation for CoLA and the mean of Pearson and Spearman correlations for STS-B. Results for RoBERTa are from Liu et al. (2019).

corpus) and ManufactuBERT$_D$ (trained on the deduplicated corpus). We tracked their downstream performance on the FabNER dataset by evaluating checkpoints saved every 500 pretraining steps. Each evaluation was performed over 10 random seeds using the fine-tuning hyperparameters described in Section 4.2.

As illustrated in Figure 2, the model trained on the deduplicated corpus converges faster. We estimate that ManufactuBERT$_D$ achieves the final performance of the baseline ManufactuBERT at approximately step 11,308. This corresponds to a 35% reduction in the required training iterations to reach the same performance level.

This accelerated convergence can be translated into energy savings. The full pretraining schedule of 17,500 steps on 8 NVIDIA V100 GPUs (250W TDP each) consumes an estimated 102,200 Wh. Reaching the equivalent performance at 11,308 steps would require only 66,032 Wh. However, this saving must be offset by the cost of the deduplication pipeline itself. As detailed in Table 6, the consumption for this process can be estimated at 2,606.3 Wh. Taking this overhead into account, the net efficiency gain from employing deduplication is approximately 32.8%.

Deduplication Stage	Time	Consumption (Wh)
MinHash Signatures	35h x 8 CPU	1 050
MinHash Buckets	1.4h x 8 CPU	42
MinHash Clusters	0.21h x 8 CPU	6,3
MinHash Filtering	0.33h x 8 CPU	10
MinHash Total		1108,3
SemDeDup Tokenizer	0.8h x 8 CPU	24
SemDeDup Embeddings	3.6h x 1 A100	1 440
SemDeDup Clustering	0.16h x 32 CPU	19,2
SemDeDup Deduplication	0.16h x 8 CPU	4,8
SemDeDup Filtering	0.33h x 8 CPU	10
SemDeDup Total		1498
Total		2606.3

Table 6: Computational cost and energy consumption of the deduplication pipeline. CPU consumption is based on Intel Cascade Lake 6248 cores (3.75W TDP per core), and GPU consumption is based on NVIDIA A100s (400W TDP).

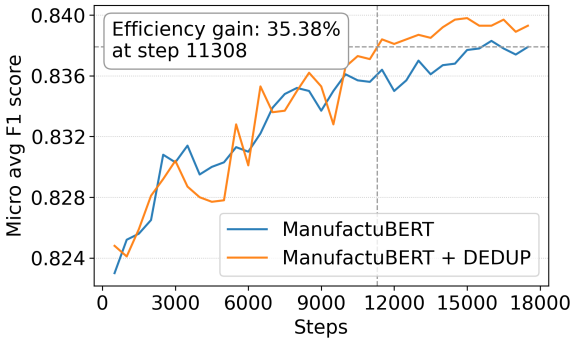

Figure 2: Performance evolution of ManufactuBERT and ManufactuBERT$_D$ on the FabNER dataset across 17,500 training steps.

5.2. Comparison of Training Costs

We compare our computational costs against those of related domain-specific models. The authors of MatSciBERT report a training time of 15 days on two V100 GPUs to adapt their model from a SciBERT checkpoint, which equates to approximately 720 V100-hours. The pretraining of SciBERT itself, conducted from scratch, required one week on a TPUv3-8. Given that this hardware is comparable to a four-V100 compute cluster[5], this training phase represents an estimated 672 V100-hours.

In contrast, our adaptation of RoBERTa required only 408 V100-hours to complete the 17,500 steps (51 hours on 8 V100s). Therefore, our methodology is not only more performant on downstream tasks but is also more computationally efficient than other specialized models in the literature.

[5] https://timdettmers.com/2018/10/17/tpus-vs-gpus-for-transformers-bert/

5.3. Effect of Deduplication Granularity

In our original SemDeDup setup, each document was represented by the average embedding of its 512-token chunks. To determine if this aggregation strategy was optimal, we investigated a more fine-grained alternative by applying SemDeDup at the chunk level. In this setup, we deduplicated the set of all 512-token chunks across the entire corpus before pretraining. We then trained a new model, ManufactuBERT$_C$, on this chunk-deduplicated dataset. We evaluated this alternative model on the FabNER benchmark. As shown in Figure 3, this more granular approach does not yield any improvement, despite requiring more computation due to the larger number of clustering points. To save computational resources, we restricted this comparative experiment to the first 12,500 pretraining steps.

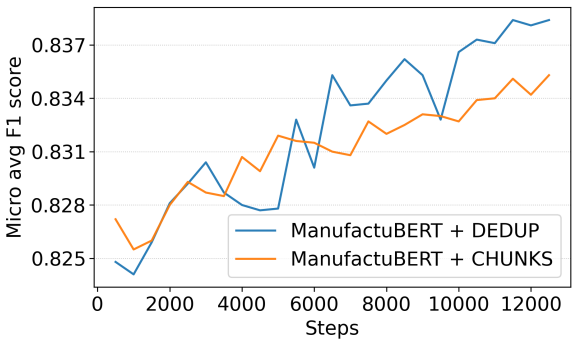

Figure 3: Performance evolution of ManufactuBERT$_C$ and ManufactuBERT$_D$ on the FabNER dataset across 12,500 training steps.

5.4. Further Deduplication Using D4

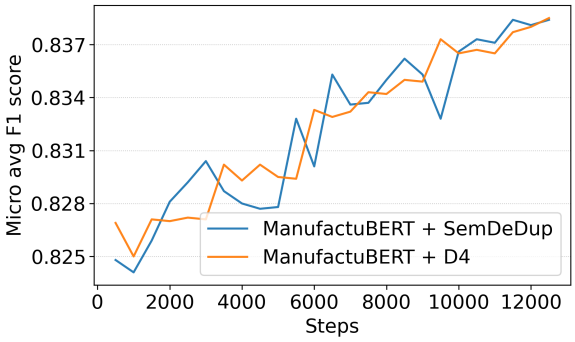

Figure 4: Performance evolution of ManufactuBERT$_{D4}$ and ManufactuBERT$_{D}$ on the FabNER dataset across 12,500 training steps.

We also investigated whether more aggressive data pruning could further improve performance. D4 (Tirumala et al., 2023) is a refinement of the SemDeDup algorithm. The authors observed that some clusters in SemDeDup are highly redundant, and that dense regions of the embedding space can degrade the quality of K-means clustering. To address this, D4 introduces two additional steps after the SemDeDup algorithm:

(1) reclustering the already deduplicated dataset

(2) retaining only the points farther to each cluster centroid according to a ratio, denoted as R_{proto}.

We applied D4 to our deduplicated dataset using R_{proto} = 0.75 and pretrained a new model, ManufactuBERT$_{D4}$. We then evaluated this model on the FabNER benchmark. As shown in Figure 4, this more aggressive deduplication strategy did not yield any additional performance gains. Moreover, D4 introduces two extra deduplication stages, increasing the overall computational cost. We also limited this comparative experiment to the first 12,500 pretraining steps.

6. Conclusion

In this work, we introduce ManufactuBERT, a RoBERTa-based language model continually pretrained on a large-scale corpus specifically curated for the manufacturing domain. By filtering the FineWeb corpus and then applying deduplication, we construct a compact, high-quality pretraining corpus.

Our empirical results validate this approach : ManufactuBERT achieves new state-of-the-art results on several manufacturing-related NLP benchmarks while maintaining strong general-domain capabilities on GLUE.

Beyond its results, our work offers a reproducible and efficient framework for domain adaptation of encoder-based language models and underscores the importance of data quality and diversity over dataset size.

Future work will explore extensions of this pipeline to other specialized domains, integration with modern encoder architectures, and alternative data selection and deduplication algorithms.

7. Limitations

Despite its strong performance, ManufactuBERT faces several limitations:

Our pretraining dataset is derived from web data and may not accurately represent documents encountered in real manufacturing contexts, which are often internal or confidential.

Our adaptation currently targets only English texts, limiting the model's applicability to multilingual or non-English manufacturing data.

Finally, although the deduplication pipeline improves data efficiency, it introduces additional preprocessing overhead, which may limit scalability for larger corpora or yield reduced benefits for smaller datasets or low-resource domains.

Acknowledgments

This publication was made possible by the use of the FactoryIA supercomputer, financially supported by the Ile-De-France Regional Council. It also benefited from the support of the DataFIX project, financed by the French government under the France 2030 Programme and operated by Bpifrance.

8. Bibliographical References

Amro Abbas, Evgenia Rusak, Kushal Tirumala, Wieland Brendel, Kamalika Chaudhuri, and Ari S. Morcos. 2024. Effective pruning of web-scale datasets based on complexity of concept clusters. In *International Conference on Learning Representations*.

Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S Morcos. 2023. Semdedup: Data-efficient learning at web-scale through semantic deduplication. *arXiv preprint arXiv:2303.09540*.

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

Margherita Bernabei, Silvia Colabianchi, and Francesco Costantino. 2022. Natural language processing applications in manufacturing: a systematic literature review.

Lola Le Breton, Quentin Fournier, Mariam El Mezouar, and Sarath Chandar. 2025. Neobert: A next-generation bert.

Andrei Z Broder. 1997. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*, pages 21–29. IEEE.

Arman Cohan, Waleed Ammar, Madeleine van Zuylen, and Field Cady. 2019. Structural scaffolds for citation intent classification in scientific publications. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3586–3596, Minneapolis, Minnesota. Association for Computational Linguistics.

Eustasio Del Barrio, Juan A Cuesta-Albertos, and Carlos Matrán. 2018. An optimal transportation approach for assessing almost stochastic order. In *The Mathematics of the Uncertain*, pages 33–44. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Rotem Dror, Segev Shlomov, and Roi Reichart. 2019. Deep dominance - how to properly compare deep neural models. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2773–2785. Association for Computational Linguistics.

Annemarie Friedrich, Heike Adel, Federico Tomazic, Johannes Hingerl, Renou Benteau, Anika Marusczyk, and Lukas Lange. 2020. The SOFC-exp corpus and neural approaches to information extraction in the materials science domain. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1255–1268, Online. Association for Computational Linguistics.

Tanishq Gupta, Mohd Zaki, NM Krishnan, et al. 2021. Matscibert: A materials domain language model for text mining and information extraction. *arXiv preprint arXiv:2109.15290*.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021a. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021b. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Jimin Hong, TaeHee Kim, Hyesu Lim, and Jaegul Choo. 2021. AVocaDo: Strategy for adapting vocabulary to downstream domain. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4692–4700, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

Yeachan Kim, Jun-Hyung Park, SungHo Kim, Juhyeong Park, Sangyun Kim, and SangKeun Lee. 2024. SEED: Semantic knowledge transfer for language model adaptation to materials science. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 421–428, Miami, Florida, US. Association for Computational Linguistics.

Martin Krallinger, Obdulia Rabal, Florian Leitner, Miguel Vazquez, David Salgado, Zhiyong Lu, Robert Leaman, Yanan Lu, Donghong Ji, Daniel M. Lowe, Roger A. Sayle, Riza Theresa Batista-Navarro, Rafal Rak, Torsten Huber, Tim Rockt"aschel, S'ergio Matos, David Campos, Buzhou Tang, Hua Xu, Tsendsuren Munkhdalai, Keun Ho Ryu, S. V. Ramanan, Senthil Nathan, Slavko Zitnik, Marko Bajec, Lutz Weber, Matthias Irmer, Saber A. Akhondi, Jan A. Kors, Shuo Xu, Xin An, Utpal Kumar Sikdar, Asif Ekbal, Masaharu Yoshioka, Thaer M. Dieb, Miji Choi, Karin Verspoor, Madian Khabsa, C. Lee Giles, Hongfang Liu, Komandur Elayavilli Ravikumar, Andre Lamurias, Francisco M. Couto, Hong-Jie Dai,

Richard Tzong-Han Tsai, Caglar Ata, Tolga Can, Anabel Usi'e, Rui Alves, Isabel Segura-Bedmar, Paloma Mart'inez, Julen Oyarzabal, and Alfonso Valencia. 2015. The chemdner corpus of chemicals and drugs and its annotation principles. *Journal of Cheminformatics*, 7(1):S2.

Aman Kumar and Binil Starly. 2022. "fabner": information extraction from manufacturing process science domain literature using named entity recognition. *J. Intell. Manuf.*, 33(8):2393–2407.

Aman Kumar, Binil Starly, and Collin Lynch. 2023. Manubert: A pretrained manufacturing science language representation model. *Available at SSRN 4375613*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Yiwei Li, Huaqin Zhao, Hanqi Jiang, Yi Pan, Zhengliang Liu, Zihao Wu, Peng Shu, Jie Tian, Tianze Yang, Shaochen Xu, Yanjun Lyu, Parker Blenk, Jacob Pence, Jason Rupram, Eliza Banu, Ninghao Liu, Linbing Wang, Wenzhan Song, Xiaoming Zhai, Kenan Song, Dajiang Zhu, Beiwen Li, Xianqiao Wang, and Tianming Liu. 2024. Large language models for manufacturing.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Zhuang Liu, Degen Huang, Kaiyu Huang, Zhuang Li, and Jun Zhao. 2020. Finbert: A pre-trained financial language representation model for financial text mining. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4513–4519. International Joint Conferences on Artificial Intelligence Organization. Special Track on AI in FinTech.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, Brussels, Belgium. Association for Computational Linguistics.

Marc Marone, Orion Weller, William Fleshman, Eugene Yang, Dawn Lawrie, and Benjamin Van Durme. 2025. mmbert: A modern multilingual encoder with annealed language learning. *arXiv preprint arXiv:2509.06888*.

Marvin Carl May, Jan Neidhöfer, Tom Körner, Louis Schäfer, and Gisela Lanza. 2022. Applying natural language processing in manufacturing. *Procedia CIRP*, 115:184–189. 10th CIRP Global Web Conference – Material Aspects of Manufacturing Processes.

Sheshera Mysore, Zachary Jensen, Edward Kim, Kevin Huang, Haw-Shiuan Chang, Emma Strubell, Jeffrey Flanigan, Andrew McCallum, and Elsa Olivetti. 2019. The materials science procedural text corpus: Annotating materials synthesis procedures with shallow semantic structures. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 56–64, Florence, Italy. Association for Computational Linguistics.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024a. The fineweb datasets: Decanting the web for the finest text data at scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Guilherme Penedo, Hynek Kydlicek, Alessandro Cappelli, Mario Sasko, and Thomas Wolf. 2024b. Datatrove: large scale data processing.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The RefinedWeb dataset for Falcon LLM: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*.

Eva Sharma, Chen Li, and Lu Wang. 2019. BIGPATENT: A large-scale dataset for abstractive and coherent summarization. *CoRR*, abs/1906.03741.

Kushal Tirumala, Daniel Simig, Armen Aghajanyan, and Ari S. Morcos. 2023. D4: Improving llm pretraining via document de-duplication and diversification.

Dennis Ulmer, Christian Hardmeier, and Jes Frellsen. 2022. deep-significance-easy and meaningful statistical significance testing in

the age of neural networks. *arXiv preprint arXiv:2204.06815*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In the Proceedings of ICLR.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. 2024. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference.

L. Weston, V. Tshitoyan, J. Dagdelen, O. Kononova, A. Trewartha, K. A. Persson, G. Ceder, and A. Jain. 2019. Named entity recognition and normalization applied to large-scale information extraction from the materials science literature. *Journal of Chemical Information and Modeling*, 59(9):3692–3702. PMID: 31361962.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.