

Assignment 1: Data Collection and Preprocessing for Foundation Model Pre-Training

1. Dataset Sources and Total Size

We collected **1.75 GB** from two public sources for domain diversity:

Source	Documents	Size	Domain	Rationale
Wikipedia (20220301.en)	200,000	835 MB	Encyclopedic	High-quality factual content (BERT, RoBERTa)
OpenWebText	200,000	952 MB	Web discussions	Conversational text (GPT-2's WebText replica)
Total	400,000	1.75 GB	Multi-domain	Formal + informal language balance

Justification: Wikipedia provides structured encyclopedic knowledge (Devlin et al., 2018), while OpenWebText offers diverse conversational web text (Radford et al., 2019). This mirrors RoBERTa's multi-source approach (Liu et al., 2019).

2. Cleaning Strategies and Reasoning

Four-Stage Pipeline:

- Format Normalization:** Removed HTML tags, markdown syntax, reference markers
 - Rationale:* Eliminates formatting noise

2. **Text Normalization:** Standardized whitespace, removed special characters (kept punctuation)
 - *Rationale:* Prevents spurious tokenization
3. **Quality Filtering:** Min 50 words, $\geq 70\%$ alphabetic, max 30% word repetition
 - *Rationale:* Filters fragments and spam
4. **Deduplication:** MD5 hashing with $O(1)$ set lookup
 - *Rationale:* Prevents overfitting to duplicates

Results: 400,000 \rightarrow 389,235 documents (**97.31% retention**) | Duplicates: 70 (0.02%) | Low-quality: 10,695 (2.67%)

The high retention rate validates our source selection, with minimal noise requiring removal.

3. Tokenization Choices

Configuration: GPT-2 BPE tokenizer | Vocab: 50,257 | Block size: 512 | Fast: Enabled (Rust backend)

Rationale: BPE handles English efficiently via subword decomposition (~ 0.75 words/token). The 512 block size balances context and computation, standard for transformers.

Chunking: Sliding windows without overlap. Example: 1,500 tokens \rightarrow Chunk 1 (1-512), Chunk 2 (513-1024), Chunk 3 (1025-1500). Minimum 50 tokens required.

Results:

Input: 389,235 documents

Output: 930,229 sequences (2.39 sequences/doc avg)

Stats: Avg length 487 tokens (95% utilization) | Total $\sim 47M$ tokens | Vocab coverage 87.3%

The 2.39 ratio indicates most documents exceed 512 tokens, typical for encyclopedic/web content.

4. Data Loader Implementation

Architecture: PyTorch with segmented storage to prevent memory issues.

SegmentedPretrainingDataset:

- *Lazy loading: Loads 1 segment (~116K sequences) at a time*
- *Memory: ~500 MB vs ~4.5 GB for full dataset (90% savings)*
- *Dynamic padding with attention masks*

DataLoader: Batch size 8 | Shuffle enabled | Workers 0 (memory-safe) | Total batches 116,279

Batch Format: `{'input_ids': [8,512], 'attention_mask': [8,512], 'labels': [8,512]}`

Key Features: Segmented storage (8 files), lazy loading with auto-switching, checkpoint recovery.

5. Challenges Encountered

Memory Exhaustion: Loading 930K sequences exceeded Colab's limit.

- *Solution: Segmented processing (8 segments), save independently, lazy load*
- *Impact: Peak memory 10 GB → 2 GB*

Tokenization Performance: Sequential processing estimated 20+ hours.

- *Solution: Batch tokenization (2K-5K docs/batch) with fast tokenizer*
- *Result: 50× speedup (25 minutes)*

Variable Sequences: Documents range 50-50,000 tokens.

- *Solution: Chunking to 512, min 50-token filter, dynamic padding*
- *Trade-off: ~5% info loss but ensures context*

Worker Crashes: Multi-process workers killed by system.

- *Solution: Single-process loading (num_workers=0)*
-

6. Reflections on Preprocessing Impact

Quality Metrics: 0.02% duplicates | 97% retention | 95% seq utilization | 87% vocab coverage | 2 domains

Training Benefits:

- **Reduced Overfitting:** Near-zero duplication, multi-domain data → generalizable patterns
- **Efficient Training:** 95% utilization → minimal wasted computation
- **Better Generalization:** Domain variety → improved zero-shot transfer

- **Meaningful Context:** 50-token minimum → long-range dependency learning

Limitations: 1.75 GB insufficient for production | English-only | 2023 data outdated

Future Work: Scale to 100 GB (Common Crawl) | Add code/papers/books |
Perplexity-based filtering | Recent data

7. Conclusion

Successfully processed 1.75 GB → 930,229 sequences with: scalable segmented architecture | 97% retention | 50× processing speedup | production-ready lazy loading.

Key Insight: Memory management via segmentation enables large-scale preprocessing within resource constraints, scalable to 100GB+ datasets.