# Training BERT-Base-Uncased to Classify Descriptive Metadata

Artem Saakov
University of Michigan
School of Information
United States
asaakov@umich.edu

***Abstract*—Libraries and archives frequently receive donor-supplied metadata in unstructured or inconsistent formats, creating backlogs in accession workflows. This paper presents a method for automating metadata field classification using a pre-trained transformer model (BERT-base-uncased). We aggregate donor metadata into a JSON corpus keyed by Dublin Core fields, flatten it into text–label pairs, and fine-tune BERT for sequence classification. On a synthetic test set spanning ten common metadata fields, we achieve an overall accuracy of 0.92. We also provide a robust inference script capable of classifying documents of arbitrary length. Our results suggest that transformer-based classifiers can substantially reduce manual effort in digital curation pipelines.**

***Index Terms*—Metadata Classification, Digital Curation, Transformer Models, BERT, Text Classification, Archival Metadata, Natural Language Processing**

## I. Introduction

Metadata underpins discovery, provenance, and preservation in digital archives. Yet many institutions face backlogs: donated items arrive faster than they can be cataloged, and donor-provided metadata—often stored in spreadsheets, text files, or embedded tags—lacks structure or consistency [1]. Manually mapping each snippet to standardized fields (e.g., Title, Date, Creator) is labor-intensive.

### *A. Project Goal*

We investigate fine-tuning Google's BERT-base-uncased model to automatically classify free-form metadata snippets into a fixed set of archival fields. By leveraging BERT's bidirectional contextual embeddings, we aim to reduce manual mapping effort and improve consistency.

### *B. Related Work*

The National Archives have explored AI for metadata tagging to improve public access [1]. Carnegie Mellon's CAMPI project used computer vision to cluster and tag photo collections in bulk [2]. MetaEnhance applied transformer models to correct ETD metadata errors with F1 $> 0.85$ on key fields [3]. Embedding-based entity resolution has harmonized heterogeneous schemas across datasets [4]. These studies demonstrate AI's potential but leave open the challenge of mapping arbitrary donor text to discrete fields.

## II. Method

### *A. Problem Formulation*

We cast metadata field mapping as single-label text classification:

- **Input:** free-form snippet $x$ (string).
- **Output:** field label $y \in \{f_1, \ldots, f_K\}$, each $f_i$ a target schema field.

### *B. Dataset Preparation*

We begin with an aggregated JSON document keyed by Dublin Core field names. A Python script (`harvest_aggregate.ipynb`) flattens this into one record per metadata entry:

```
{"text":"Acquired on 12/31/2024","label":"Date"}
```

Synthetic expansion to 200 examples across ten fields ensures coverage of varied formats.

### *C. Model Fine-Tuning*

- **Model:** `bert-base-uncased` with $K = 10$ labels.
- **Tokenizer:** WordPiece, padding/truncation to 128 tokens.
- **Training:** 80/20 split, cross-entropy loss, LR=2e-5, batch size=8, 5 epochs via Hugging Face `Trainer` [5].
- **Evaluation:** Accuracy, weighted and macro F1, precision, and recall using the `evaluate` library.

### *D. Inference Pipeline*

We package our inference logic in `bertley.py`. It loads the fine-tuned model, tokenizes input (text or file), and handles documents longer than 512 tokens by chunking with overlap (stride=50). Pseudocode excerpt:

```
# Load model & tokenizer from checkpoint
tokenizer = AutoTokenizer.from_pretrained(model_di
model = AutoModelForSequenceClassification.from_pr
classifier = pipeline("text-classification",
                      model=model,
                      tokenizer=tokenizer,
                      return_all_scores=True)


# For long texts, split into overlapping chunks
def chunk_and_classify(text):
  tokens = tokenizer(text)['input_ids'][0]
  for i in range(0, len(tokens), max_len - stride):
    chunk = tokenizer.decode(tokens[i:i+max_len])
    scores = classifier(chunk)
    accumulate(scores)
  return average_scores()
```

This script achieves robust, batch-ready inference for entire documents.

## III. Results

### A. Evaluation Metrics

After fine-tuning for 5 epochs, we evaluated on the test set. Table I summarizes the results:

TABLE I
Test Set Evaluation Metrics

| Metric | Value |
|---|---|
| Loss | 0.1338 |
| Accuracy | 0.9665 |
| F1 (weighted) | 0.9628 |
| Precision (weighted) | 0.9650 |
| Recall (weighted) | 0.9665 |
| F1 (macro) | 0.8283 |
| Precision (macro) | 0.8551 |
| Recall (macro) | 0.8225 |
| Runtime (s) | 35.83 |
| Samples/sec | 518.70 |
| Steps/sec | 16.22 |

### B. Interpretation

Overall accuracy of 96.65% and weighted F1 of 96.28% demonstrate reliable field mapping. The macro F1 (82.83%) suggests room for improvement on rarer or more ambiguous classes. Inference speed ( 100 snippets/s on GPU) is sufficient for large-scale backlog processing.

## IV. Conclusion

Fine-tuning BERT-base-uncased for metadata classification yields an overall accuracy of 0.92, confirming the viability of transformer-based automation in digital curation. Future work will integrate real EAD finding aids, implement multi-label classification for ambiguous entries, and incorporate human-in-the-loop validation.

## Acknowledgment

The author thanks the University of Michigan School of Information and participating archival staff for insights into donor metadata workflows.

## References

[1] U.S. National Archives and Records Administration, “Artificial intelligence at the National Archives.” [Online]. Available: https://www.archives.gov/ai, accessed Apr. 4, 2025.

[2] Carnegie Mellon Univ. Libraries, “Computer vision archive helps streamline metadata tagging,” Oct. 2020. [Online]. Available: https://www.cmu.edu/news/stories/archives/2020/october/computer-vision-archive.html.

[3] M. H. Choudhury *et al.*, “MetaEnhance: Metadata Quality Improvement for Electronic Theses and Dissertations,” *arXiv*, Mar. 2023.

[4] K. Sawarkar and M. Kodati, “Automated metadata harmonization using entity resolution & contextual embedding,” *arXiv*, Oct. 2020.

[5] T. Wolf *et al.*, “HuggingFace Transformers: State-of-the-art natural language processing,” in *Proc. EMNLP: Findings*, 2020, pp. 8201–8210.