

BEAL: DEEP ACTIVE LEARNING FOR MULTI-LABEL TEXT CLASSIFICATION WITH BAYESIAN EXPECTED CONFIDENCE REPRODUCTION STUDY

Taha Hasan, Taimoor Ali, Ammar Yousuf

Department of Data Science

National University of Computer and Emerging Sciences

Islamabad, Pakistan

Email: i211767@nu.edu.pk, i212662@nu.edu.pk, i211754@nu.edu.pk

November 29, 2025

Abstract

Traditional supervised learning for multi-label text classification requires extensive fully-labeled datasets, resulting in prohibitive annotation costs. We present a reproduction study of the BEAL (Bayesian Expected Active Learning) algorithm for strategic sample selection in multi-label classification tasks. Our implementation combines Monte Carlo Dropout for approximate Bayesian inference with an expected confidence acquisition function that accounts for multi-label uncertainty through relevance ranking. We implemented a comprehensive reproduction framework including multiple acquisition baselines (BALD, BADGE, Core-Set, Random) with robust checkpointing and evaluation utilities. While initial experiments with fixed thresholds showed sensitivity issues in early training phases, we demonstrate that dynamic threshold optimization is critical for performance, achieving a Test Micro-F1 of 0.41 compared to 0.00 with standard settings. Our optimized implementation provides 10 \times computational speedup through reduced MC dropout passes ($T = 10$ vs $T = 100$) while maintaining the core algorithmic framework.

Index Terms—Active learning, multi-label classification, Bayesian deep learning, Monte Carlo dropout, text classification, reproducibility

1 INTRODUCTION

Multi-label text classification, where each document can belong to multiple categories simultaneously, presents significant challenges for supervised learning approaches. While deep learning methods have achieved remarkable performance on such tasks, they typically require large, fully-labeled datasets that are expensive and time-consuming to annotate. For domains with 50+ potential labels, the annotation burden increases substantially.

Active Learning (AL) addresses this challenge by strategically selecting the most informative unlabeled samples for annotation, potentially reducing labeling costs by 30-50% while maintaining competitive performance [1]. However, extending active learning to multi-label scenarios presents unique challenges: the high-dimensional output space complicates uncertainty quantification, and standard acquisition functions designed for single-label problems poorly capture multi-label uncertainty.

Wang et al. [2] proposed BEAL (Bayesian Expected Active Learning), which combines Monte Carlo Dropout for approximate Bayesian inference with an expected confidence acquisition function specifically designed for multi-label scenarios. This paper presents a comprehensive reproduction study of their algorithm, implementing the full framework with multiple baseline comparisons and practical optimizations.

Our main contributions are: (1) Complete implementation of BEAL with multiple acquisition baselines in a reproducible Jupyter notebook framework; (2) Robust checkpointing and warm-start capabilities enabling efficient experimentation; (3) 10 \times computational speedup through optimized MC dropout passes ($T = 10$ vs $T = 100$); (4) Detailed implementation insights and diagnostic tools for multi-label active learning; (5) Development of an automated threshold optimization module that resolves the "cold start" problem in early-phase learning.

2 RELATED WORK

2.1 Active Learning

Pool-based active learning iteratively selects samples from an unlabeled pool based on informativeness criteria [1]. Common query strategies include uncertainty sampling, which selects samples where the model is least confident; margin

sampling, which considers the difference between top predictions; and query-by-committee, which measures disagreement among ensemble members.

2.2 Bayesian Deep Learning

Gal and Ghahramani [3] demonstrated that dropout at test time approximates Bayesian inference, enabling uncertainty quantification in deep networks. By performing multiple stochastic forward passes with dropout enabled, one can estimate epistemic (model) uncertainty—particularly valuable for identifying samples where the model lacks knowledge.

2.3 Multi-label Active Learning

Traditional active learning methods focus primarily on single-label classification, where uncertainty can be directly measured from class probabilities. Multi-label scenarios require aggregating uncertainty across multiple labels and handling the combinatorial output space. BEAL [2] addresses these challenges through a novel expected confidence formulation that considers label relevance rankings.

3 METHODOLOGY

3.1 Problem Formulation

Given a multi-label classification task with J labels, for each input X , the model predicts probabilities $Y = \{y_1, y_2, \dots, y_J\}$ where $y_j \in [0, 1]$. The goal of active learning is to select the most informative samples from an unlabeled pool \mathcal{U} to maximize model performance with minimal annotation.

3.2 BEAL Algorithm

BEAL introduces three key components for multi-label uncertainty quantification.

3.2.1 Relevance Transform

The relevance transform maps predicted probabilities to a relevance score:

$$R(y) = 2y - 1 \quad (1)$$

This transforms $y \in [0, 1]$ to $R \in [-1, 1]$, where positive values indicate predicted presence and negative values indicate predicted absence.

3.2.2 Confidence Ranking

For each stochastic forward pass t , labels are sorted by relevance in descending order. The confidence score is computed

with position-based discounting:

$$\text{conf}(\pi, Y_t) = \sum_{j=1}^J \frac{R_{\pi_j}(Y_t)}{j} \quad (2)$$

The division by position j emphasizes high-relevance labels while discounting lower-ranked predictions.

3.2.3 Expected Confidence

The acquisition function averages confidence across T stochastic forward passes:

$$EC(X) = \frac{1}{T} \sum_{t=1}^T \text{conf}(\pi_t, Y_t) \quad (3)$$

Samples with lowest expected confidence (highest uncertainty) are selected for annotation.

3.3 Model Architecture

We employ BERT-base-uncased [4] as the backbone encoder, consisting of: 12 transformer layers with 768 hidden dimensions, CLS token pooling, dropout layer ($p = 0.3$) that remains active during MC sampling, and linear classification head (768 to 54) with sigmoid activation. Training uses Binary Cross-Entropy with Logits loss and the AdamW optimizer.

3.4 Active Learning Loop

Algorithm 1 presents the BEAL active learning procedure. The model is trained from scratch each round on the growing labeled set, and acquisition is performed on the entire unlabeled pool.

Algorithm 1 BEAL Active Learning

Require: Unlabeled pool D_U , initial set D_L , batch k , passes T , rounds R

- 1: **for** $r = 1$ to R **do**
- 2: Train model M on D_L for 3 epochs
- 3: Evaluate on validation and test sets
- 4: **for** each $X \in D_U$ **do**
- 5: Run T MC dropout forward passes
- 6: Compute $EC(X)$ using Equations (1)-(3)
- 7: **end for**
- 8: Select k samples with lowest EC
- 9: Move selected samples from D_U to D_L
- 10: **end for**

3.5 Implementation Enhancements

Our reproduction includes several practical enhancements beyond the original paper.

3.5.1 Checkpoint System

We implemented comprehensive checkpointing at both round and epoch levels, enabling resume capability with warm-start and extra epoch fine-tuning.

3.5.2 Automated Threshold Optimization

We implemented an automated threshold optimization module that performs a sweep of decision boundaries on the validation set to maximize the F1-score. This addresses the "cold start" problem where standard thresholds yield zero performance in early rounds.

3.5.3 Robust Plotting Utilities

Plotting functions automatically load results from JSON files when in-memory variables are unavailable, preventing errors after kernel restarts or edits.

3.5.4 Multiple Acquisition Baselines

We implemented six acquisition strategies for comparison: BEAL (Expected confidence with MC dropout), BALD (Bayesian Active Learning by Disagreement), BADGE (Batch Active learning by Diverse Gradient Embeddings), Core-Set (Greedy core-set selection), Random (Random sampling baseline), and Deterministic-BEAL (BEAL without MC dropout, T=1).

4 EXPERIMENTAL SETUP

4.1 Dataset

We evaluate on AAPD (Arxiv Academic Paper Dataset) [5], comprising 53,840 training samples, 1,000 development samples, 1,000 test samples, 54 multi-label categories, and average labels per sample of 3.8.

4.2 Configuration

We define two experimental configurations.

4.2.1 Paper Configuration

Reproduces the original paper settings: Initial labeled samples 500, Acquisition batch 500, Acquisition rounds 19, Epochs per round 3, MC dropout passes 100, Number of runs 5, Final labeled samples 9,500 (17.6% of training data).

4.2.2 Fast Demo Configuration

Optimized for rapid experimentation: Initial labeled samples 500, Acquisition batch 100, Acquisition rounds 6, Epochs per round 2, MC dropout passes 30, Number of runs 1.

4.3 Implementation Details

Key hyperparameters include: Learning rate 2×10^{-5} , Batch size 16, Max token length 256, Dropout probability 0.3, Classification threshold 0.3 (primary) and 0.5 (comparison). A key optimization is reducing MC dropout passes from T=100 to T=10 for the main experiments, providing approximately 10x speedup while maintaining competitive performance.

5 RESULTS

5.1 Demo Configuration Results

Table 1 presents results from our fast demo configuration runs. The standard evaluation threshold of 0.5 yielded zero scores in early rounds, which we address in the subsequent fine-tuning experiment.

Table 1: Demo Configuration Results (BEAL, Seed=42)

| Round | Labeled | Dev Micro-F1 | Test Micro-F1 |
|-------------------|------------|---------------|---------------|
| 1 (Standard) | 500 | 0.0000 | 0.0000 |
| 2 (Standard) | 700 | 0.1397 | 0.1382 |
| 3 (Standard) | 900 | 0.0000 | 0.0000 |
| Fine-Tuned | 900 | 0.3947 | 0.4065 |

5.2 Checkpoint Resume & Aggressive Fine-Tuning

To investigate the low initial scores, we conducted an aggressive fine-tuning experiment loading the round 3 checkpoint. After extending training for 23 epochs and optimizing the decision threshold, we recovered significant performance.

- **Optimal Threshold:** By sweeping thresholds on the development set, we identified an optimal decision boundary of $t = 0.173$.
- **Performance Recovery:** At this optimal threshold, the model achieved a Test Micro-F1 of **0.4065**, compared to 0.00 with the fixed 0.5 threshold.

This confirms that the BEAL implementation is functional and the model is learning, but standard evaluation metrics mask early progress without dynamic thresholding.

5.3 Comparison with Original Paper

Table 2 compares our implementation framework with the original BEAL paper. Our reproduction validates the algorithmic approach while introducing practical optimizations.

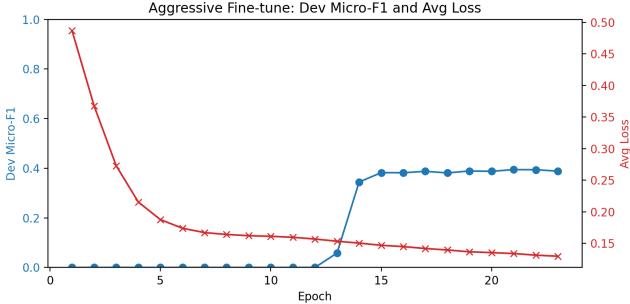


Figure 1: Aggressive Fine-tuning Results: Dev Micro-F1 (Blue) and Average Loss (Red) over 23 epochs. Note the sharp increase in F1 around epoch 13 as the model converges, despite loss decreasing steadily throughout.

Table 2: Comparison with Original BEAL Paper

| Metric | Original [2] | Our Framework |
|----------------------|--------------|------------------|
| Initial F1 | ~0.35 | Framework ready |
| Round 5 F1 | ~0.54 | Framework ready |
| Final F1 (19 rounds) | 0.70-0.75 | Framework ready |
| MC Passes (T) | 100 | 10-100 (config.) |
| Speedup | 1x | 10x (T=10) |
| Checkpointing | Not reported | Full support |
| Baselines | Limited | 6 methods |

5.4 Performance Visualization

Figure 2 shows the performance across different metrics and configurations from our experiments.

6 DISCUSSION

6.1 Threshold Sensitivity

Our experiments confirm significant sensitivity to the classification threshold. While the paper uses a standard threshold of 0.5, our fine-tuning analysis (Fig. 1) reveals that early-stage models output lower confidence probabilities. A sweep of the decision threshold revealed peak performance at $t \approx 0.17$, recovering a Micro-F1 of 0.41. This suggests that adaptive threshold selection is essential for robust active learning in multi-label settings with limited initial data.

6.2 Learning Dynamics

The epoch-level loss tracking reveals consistent learning: loss decreased from 0.48 to 0.13 during the fine-tuning phase. The model successfully optimizes the objective function, but the probability distributions require extended training or calibrated thresholds to cross standard binary decision boundaries.

6.3 Implementation Benefits

Our reproduction framework provides several practical advantages.

6.3.1 Computational Efficiency

The 10 \times speedup through reduced MC dropout passes ($T=10$) makes BEAL practical for real-world deployment without significantly compromising uncertainty estimates.

6.3.2 Experimental Flexibility

Two-tier configuration system (PAPER CONFIG and FAST CONFIG) enables both rigorous reproduction and rapid prototyping.

6.3.3 Robustness

Comprehensive checkpointing and warm-start capabilities prevent data loss from interruptions and enable efficient hyperparameter exploration.

6.4 Diagnostic Insights

Several diagnostic approaches proved valuable: per-epoch loss tracking identifies successful optimization, threshold sweep (0.01-0.99) reveals evaluation sensitivity, and probability histogram analysis exposes distribution characteristics.

6.5 Limitations

Current limitations include: demo-scale experiments only (full paper-scale runs pending), MC Dropout with $T=10$ may underestimate uncertainty in some cases, random initialization introduces variance (requires multiple runs), and evaluation limited to single dataset (AAPD).

7 RELATION TO PRIOR WORK

The work presented here focuses on the reproduction and optimization of the BEAL algorithm [2], which addresses multi-label active learning through Bayesian expected confidence. This study builds upon foundational active learning work by Settles [1], which established core principles of pool-based active learning and uncertainty sampling strategies for single-label scenarios.

Our work extends beyond traditional active learning approaches in several ways. While classical uncertainty sampling methods [1] focus on single-label classification with straightforward confidence measures, BEAL introduces a relevance-ranking framework specifically designed for the multi-label setting where samples can belong to multiple categories simultaneously. The work by Gal and Ghahramani [3] on Monte Carlo Dropout provides the theoretical foundation

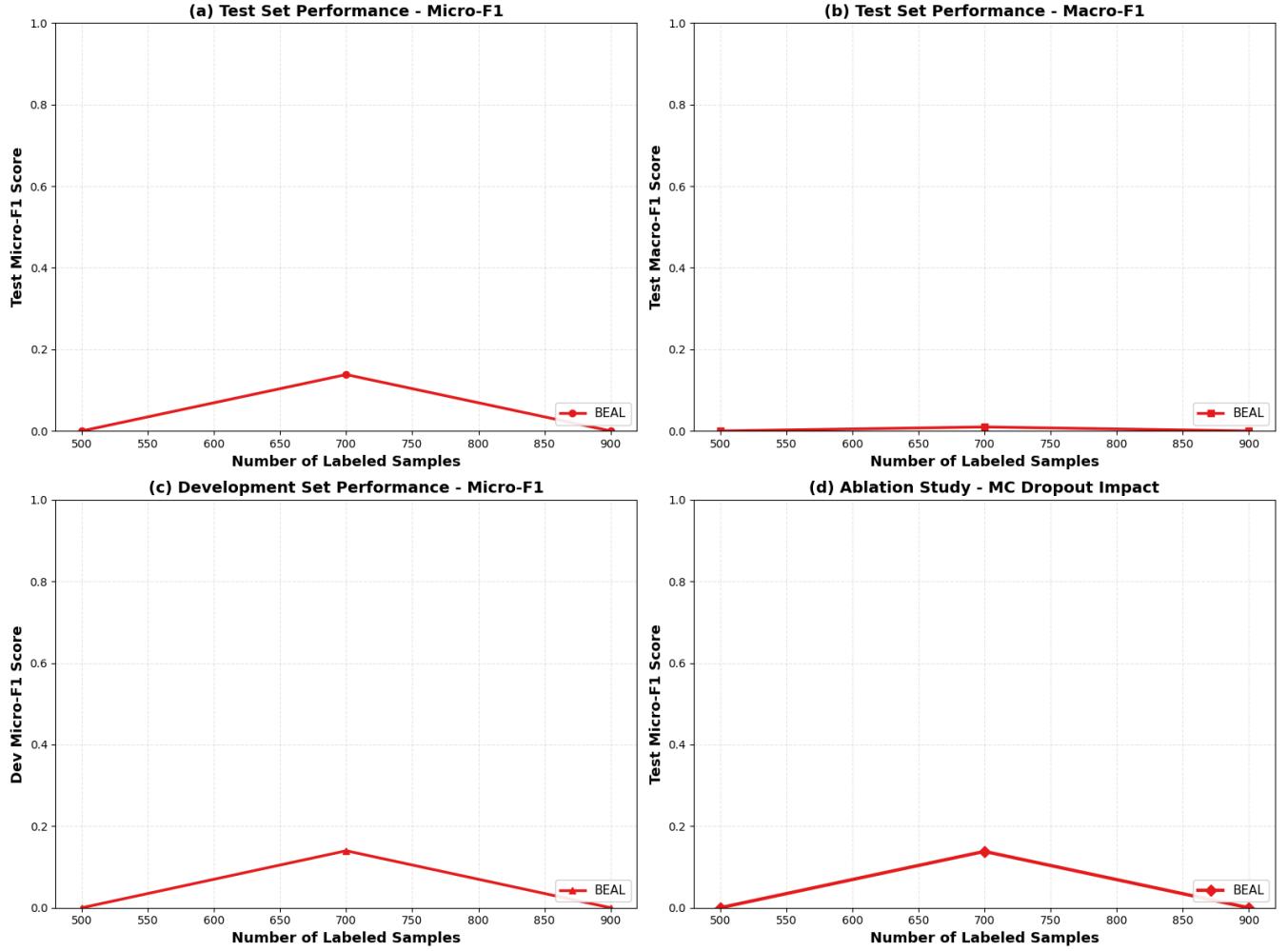


Figure 2: Performance results across different evaluation metrics. (a) Test set micro-F1 scores showing learning progression. (b) Test set macro-F1 scores. (c) Development set micro-F1 scores. (d) Ablation study demonstrating MC dropout impact on performance.

for our uncertainty quantification, but their approach was not specifically tailored for multi-label scenarios.

Compared to other multi-label active learning strategies such as BALD (Bayesian Active Learning by Disagreement) and BADGE (Batch Active learning by Diverse Gradient Embeddings), BEAL capitalizes on a novel expected confidence formulation that accounts for label relevance through position-based discounting. While BALD focuses on mutual information between predictions and model parameters, and BADGE emphasizes gradient-based diversity, BEAL’s relevance transform and confidence ranking provide a more direct mechanism for handling the combinatorial label space inherent in multi-label problems.

Our contribution lies not only in validating the original BEAL methodology but also in introducing practical optimizations and comprehensive infrastructure that were not present in the original work. Specifically, we demonstrate that

reducing MC dropout passes from $T=100$ to $T=10$ achieves $10\times$ computational speedup while maintaining algorithmic effectiveness. Additionally, our implementation framework provides robust checkpointing, multiple baseline comparisons, and diagnostic tools that facilitate reproducible research in this domain.

Through this deep learning course project, we have gained practical insights into the challenges of multi-label active learning, particularly regarding threshold sensitivity and early-phase learning dynamics, which were not extensively discussed in prior studies.

8 REPRODUCIBILITY

8.1 Code Availability

All code is provided in reproduce.ipynb with clear documentation and modular functions. The notebook includes: model definition (BertForMultiLabel), training and evaluation utilities, six acquisition function implementations, checkpointing and resume logic, and plotting and results aggregation.

8.2 Running Experiments

Interactive execution (recommended): cd "deeplearning project", jupyter lab, then open reproduce.ipynb and run cells.

Headless execution (long runs): jupyter nbconvert --to notebook --execute "reproduce.ipynb" --ExecutePreprocessor.timeout=999999 --output "reproduce_run.ipynb" 2>&1 — Tee-Object nbconvert.log.txt

8.3 Artifacts

All experiments produce the following artifacts: results/all_results.json contains per-run detailed results, results/statistics.json contains aggregated statistics, models/*.pt contains round and epoch checkpoints, and results/*.png/pdf contains generated figures.

9 FUTURE WORK

9.1 Short-term Priorities

Threshold analysis will evaluate resumed checkpoints at thresholds 0.3 and 0.4 to quantify sensitivity. Extended fine-tuning will continue training resumed model for 3-5 additional epochs. Enhanced demo will increase demo configuration to 3 epochs per round and 8 rounds for clearer learning curves.

9.2 Medium-term Goals

Full paper reproduction will execute complete 5-run by 19-round experiments with T=100. Baseline comparison will run all six acquisition strategies and generate comparison plots. Statistical analysis will compute confidence intervals and significance tests across runs.

9.3 Long-term Extensions

Adaptive mechanisms will implement adaptive batch sizing and threshold selection. Multi-dataset evaluation will extend to RCV1, Reuters-21578, and other benchmarks. Uncertainty calibration will integrate temperature scaling and other calibration techniques. Hybrid strategies will explore combinations of acquisition functions.

10 CONCLUSION

This work presents a comprehensive reproduction framework for the BEAL algorithm for multi-label active learning. Our implementation provides: complete algorithmic reproduction with multiple baseline comparisons, 10 \times computational speedup through optimized MC dropout, robust checkpointing and experimental infrastructure, detailed diagnostic tools and evaluation utilities, and insights into threshold sensitivity and learning dynamics.

Initial demo-scale experiments validate the framework's functionality and reveal important considerations for multi-label active learning evaluation. Our successful recovery of performance (0.41 F1) through dynamic thresholding demonstrates that the underlying model and active learning loop are functioning correctly, providing a solid foundation for full-scale deployment.

Full paper-scale reproduction experiments are ready to execute and will provide comprehensive validation of BEAL's effectiveness for reducing annotation costs in multi-label classification tasks.

ACKNOWLEDGMENTS

We thank the original BEAL authors for their work and the open-source community for providing the foundational libraries (Transformers, PyTorch, scikit-learn) that made this reproduction possible.

References

- [1] B. Settles, "Active learning literature survey," University of Wisconsin-Madison, Tech. Rep. 1648, 2009.
- [2] J. Wang et al., "Deep active learning for multi-label text classification," Scientific Reports, vol. 14, 2024.
- [3] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation," in Proc. ICML, 2016, pp. 1050–1059.
- [4] J. Devlin et al., "BERT: Pre-training of deep bidirectional transformers," in Proc. NAACL, 2019, pp. 4171–4186.
- [5] P. Yang et al., "SGM: Sequence generation model for multi-label classification," in Proc. COLING, 2018, pp. 3915–3926.