LESS IS MORE: ADAPTIVE COVERAGE FOR SYNTHETIC TRAINING DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Models (LLMs) enable rapid generation of synthetic training data for downstream classifiers, offering a solution when human-labeled data is costly, scarce, or time-sensitive. However, synthetic datasets suffer from systematic redundancy: LLMs over-generate common patterns while under representing nuanced edge cases, leading to training inefficiency and degraded generalization. We introduce Adaptive Coverage Sampling (ACS), a principled method that formulates synthetic data selection as a graph-based maximum coverage problem over semantic similarity. By constructing a similarity graph with adaptively tuned thresholds and applying greedy approximation, ACS identifies maximally diverse, representative subsets without requiring iterative model training or expensive quality scoring. We demonstrate a striking "less is more" phenomenon across sentiment analysis, relation extraction, and named entity recognition tasks: classifiers trained on ACS-selected subsets comprising just 10-30% of the original synthetic data match or exceed the performance of models trained on full datasets. This dramatic data reduction translate directly to computational savings in fine-tuning costs while improving model generalization through enhanced diversity. Our results establish that carefully curated synthetic data systematically outperforms naive utilization of large, redundant corpora, and that intelligent subset selection is essential for effective synthetic data utilization.

1 INTRODUCTION

The emergence of highly capable Large Language Models (LLMs) such as OpenAI's GPT (Achiam et al., 2023) and Google's Gemini (Team et al., 2023; 2024; 2025) has fundamentally transformed how we approach the longstanding challenge of obtaining training data for machine learning systems. In domains where human annotation is prohibitively expensive (e.g. specialized medical or legal classifications), or where rapid deployment is critical (e.g. emerging online media trends), the ability to generate synthetic labeled data on demand offers unprecedented flexibility (Bunte et al., 2021; Ding et al., 2022; Meng et al., 2022). This capability has sparked widespread interest in using LLM-generated data to train smaller, deployable models for various natural language tasks (Li et al., 2024; Kuo et al., 2024).

However, the abundance of synthetic data does not guarantee superior model performance. Recent studies reveal a critical paradox: while synthetic datasets can grow arbitrarily large at minimal cost, naively training on these corpora often produces models that under perform compared to those trained on carefully curated subsets (Gandhi et al., 2024; Liu et al., 2024; Long et al., 2024). The core issues lies in systematic redundancy and imbalance inherent to LLM generation. When prompted to produce training examples, LLMs exhibit strong biases toward frequent, prototypical patterns while underrepresenting nuanced edge cases. Consider hate speech detection, where distinguishing between explicit toxicity, subtle sarcasm, coded language, and context-dependent harm is crucial. An LLM may generate hundreds of straightforwardly offensive examples but only a handful representing borderline cases or indirect forms of harm (Gandhi et al., 2024; Hao et al., 2024). This skewed representation not only dilutes the informative content of synthetic datasets but actively degrades model generalization by over-saturating training on common patterns while starving the model of exposure to critical minority cases (Kuo et al., 2024).

The problem extends beyond simple class imbalance. Even within a single class, LLMs produce semantically repetitive samples that offer diminishing marginal information (Li et al., 2024). Training on these redundant examples wastes computational resources during fine-tuning and can lead to overfitting on the particular phrasings and structures favored by the generator model, rather than learning robust decision boundaries (Toneva et al., 2018; Paul et al., 2021; Sorscher et al., 2022). Moreover, as synthetic datasets scale to tens of thousands of examples, the computational cost of fine-tuning grows linearly, motivating the question: *can we achieve comparable or superior performance using only a carefully selected subset of the synthetic data?*

The Knowledge Distillation Perspective. Our work addresses this question within the broader paradigm of knowledge distillation and model compression. In practical deployment scenarios, large LLMs serve as expensive "teacher" models that possess broad capabilities but are too costly, slow, or sensitive for direct production use. The standard solution involves using the teacher to generate synthetic training data, which is then used to fine-tune smaller, efficient "student" (or specialist) models that can meet latency, cost and privacy constraints (Gou et al., 2021). This pipeline is ubiquitous in industry applications where inference speed and resource efficiency are paramount. Within this framework, the quality and efficiency of the synthetic training set directly determines both the performance of the student model and the computational cost of its training. Our central thesis is that intelligent data selection can simultaneously improve student model accuracy and reduce training costs by identifying the most informative, diverse subset of the synthetic corpus.

Our Contribution: Adaptive Coverage Sampling. We propose **Adaptive Coverage Sampling (ACS)**, a principled approach to synthetic data curation that frames subset selection as a graph-based maximum coverage problem. Unlike heuristic methods that score individual samples based on training dynamics (Toneva et al., 2018; Paul et al., 2021) or require expensive LLM-based quality rating (Chen et al., 2023), ACS operates on the geometric structure of the data itself. The algorithm is as follows:

1. **Semantic Embedding:** Synthetic text samples are embedded into a latent space where semantic similarity can be measured via cosine similarity.
2. **Graph Construction:** Samples become nodes in a similarity graph, with edges connecting samples whose similarity exceed a threshold, τ. A degree constraint $d_{\max}$ limits each node's neighbors, preventing over-connection to generic hub samples and promoting local diversity.
3. **Adaptive Threshold Tuning:** We conduct a binary search over τ to find the sparsest graph that achieves a target coverage level, c (the fraction of samples either selected or adjacent to a selected sample). This search is theoretically justified by our proof of coverage monotonicity (Section 2).
4. **Maximum Coverage:** Given the optimally pruned graph, we iteratively select k samples that collectively cover the largest portion of the dataset, using the classical greedy approximation with $(1 - 1/e)$ optimality guarantee.

On the theoretical side, we prove (Theorem 2.2) that for the exact maximum coverage problem, coverage is monotonic in the similarity threshold, validating our binary search procedure. We empirically verify that this monotonicity persists under greedy approximation. For scalability, we prove (Theorem D.1) that the optimal threshold can be estimated on a random subsample with bounded error via a Hoeffding-type concentration inequality, enabling application to large-scale datasets without exacerbating the computational overhead of pairwise similarities.

Empirically, we evaluate ACS across three diverse NLP tasks with varying difficulty: binary sentiment classification (SST2 (Socher et al., 2013)), relation extraction (FewRel (Han et al., 2018)), and Named Entity Recognition (CrossNER (Liu et al., 2021)). In all cases, models fine-tuned on ACS-selected subsets comprising just 10-30% of the synthetic data match or outperform models trained on the full dataset. Critically, these compact subsets exhibit significantly higher diversity (measured by inverse SelfBLEU scores) than competing methods, directly correlating with improved generalization.

We compare ACS against both classical data selection techniques (random sampling, EL2N scoring (Paul et al., 2021), forgetting scores (Toneva et al., 2018), prototypicality (Sorscher et al., 2022))

and modern LLM-based filtering (AlpaGasus (Chen et al., 2023)). ACS consistently outperforms all baselines, particularly at aggressive pruning rates where the benefits of identifying diverse, representative samples are most pronounced. We further observe that moderate coverage levels ($c = 0.9$) consistently outperform full coverage, suggesting that excluding the most redundant or potentially noisy samples improves downstream performance.

By achieving equivalent performance with 70–90% less training data, ACS proportionally reduces fine-tuning costs in GPU hours, energy consumption, and carbon footprint. These savings compound significantly for organizations deploying multiple task-specific models or frequently retraining on updated synthetic corpora. Beyond these efficiency gains, our work establishes a fundamental principle for synthetic data utilization: quality, measured by diversity and representativeness, systematically outperforms quantity, challenging the default practice of training on all available synthetic data.

2 Preliminaries & Methodology

In this section, we present our pipeline for selecting representative subsets from large synthetic datasets. We begin by describing the synthetic data and baseline methods used for comparison, then introduce ACS, emphasizing both its theoretical foundations and its flexibility for scalable implementation. ACS is a general framework: while we present specific instantiations (Gecko embeddings, exact pairwise similarities), each component can be substituted with more efficient approximations as needed for different scales and domains. We here note that due to space constrains, we defer our discussion of how ACS situates itself within the literature on selection methods and synthetic data to Appendix A.

2.1 Synthetic Data Generation

We utilize synthetic corpora generated by GPT-3.5 (Achiam et al., 2023) using established prompt templates tailored to specific downstream tasks following prior work (Ding et al., 2022). Each dataset is balanced across labels to ensure sufficient initial diversity. As documented in recent studies (Long et al., 2024), LLM-generated datasets exhibit systematic redundance: similar semantic content is generated repeatedly, with common patterns over-represented and nuanced edge cases under-represented. ACS targets precisely this redundancy, extracting informative and diverse subsets that preserve representational coverage while eliminating repetitive samples.

2.2 Downsampling Methods

Our goal is to select a subset of size $k < N$ from an initial corpus of size N, maximizing diversity and representativeness to improve downstream model training efficiency.

Baseline Methods. We benchmark ACS against widely used data selection techniques, each representing a distinct selection paradigm.

Random sampling selects k samples uniformly at random, serving as a naive baseline. **k-Means** clusters embeddings into k centroids and selects points nearest each center, promoting coarse-grained diversity across the semantic space. **SemDeDup** (Abbas et al., 2023) removes near-duplicate samples under a similarity threshold, explicitly targeting redundancy before downstream selection. **EL2N** (Paul et al., 2021) ranks samples by average L_2 distance between model predictions and true labels across early training checkpoints, prioritizing persistently challenging examples. **Forgetting scores** (Toneva et al., 2018) count transitions between correct and incorrect predictions during training, emphasizing samples near decision boundaries. **Prototypicality** (Sorscher et al., 2022) computes class-specific centroids in embedding space and prioritizes samples closest to these centroids, selecting representative class examples. **LLM rater (AlpaGasus)** (Chen et al., 2023) employs GPT-3.5 to assign quality ratings to each synthetic input-output pair, retaining only the highest-ranked samples (in the present experiments we use Gemini-2.5-flash as the rater).

Each baseline ranks the dataset according to its respective criterion and selects the top k samples. Note that EL2N and forgetting scores require multiple training runs to compute, while prototypicality and ACS operate directly on embeddings without model training.

Adaptive Coverage Sampling. ACS formulates subset selection as a graph-based maximum coverage problem, providing a principled approach to balancing diversity and representativeness. The algorithm consists of four key steps.

Step 1: Semantic Embedding. Samples are embedded into a latent space where semantic similarity can be measured. In our experiments, we use Gecko embeddings (Lee et al., 2024), though ACS is compatible with any embedding method. The choice of embedding determines what "similarity" means in the application context.

Step 2: Graph Construction. We construct a similarity graph $G = (V, E)$ where each sample is a node in V, and edges connect samples whose cosine similarity exceeds a threshold τ. To promote diversity and prevent over-connection to generic samples, we impose a degree constraint: each node connects to at most $d_{\max}$ neighbors, selecting those with highest similarity. This constraint, derived from the pigeonhole principle as $d_{\max} > {}^{c \cdot N}/_{k}$ for target coverage c, ensures sufficient connectivity while forcing the algorithm to select a distributed set of representatives rather than a few central points.

As an implementation note, for large-scale datasets, graph construction can be accelerated using approximate nearest neighbor methods (e.g. Locality-Sensitive Hashing (Chen et al., 2022; Shekkizhar et al., 2023)) or by computing similarities only within partitions. These approximations preserve the essential coverage structure while reducing computational cost from $O(N^2)$ to near-linear complexity.

Step 3: Adaptive Threshold Tuning. The threshold τ controls graph sparsity and thus semantic resolution: high thresholds yield disconnected graphs (near random selection), while low thresholds yield dense graphs (selecting only a few central nodes). We automatically determine the optimal threshold via binary search, targeting a user-specified coverage level. Coverage quantifies representational breadth.

Definition 2.1 (Coverage)**.** *Let $G = (V, E)$ be a graph with vertex set V, edge set E, and self-loop for all vertices. A subset $H \subseteq V$ of size $|H| = k$ achieves coverage $c \in [0, 1]$ if*

$$\left| \bigcup_{i \in H} N_i \right| = c \cdot |V|$$

where N_i is the neighborhood of vertex $i \in H$ (ie. i covers the elements of N_i, including itself).

Coverage of 1.0 means every sample is either selected or adjacent to a selected sample, while lower coverage strategically excludes the most redundant or potentially noisy samples. Empirically, we find that moderate coverage ($c \approx 0.9$) often outperforms full coverage, suggesting that the least-connected samples may represent outliers or low-quality generations (see Section 3). The binary search for thresholding is justified by the following monotonicity property.

Theorem 2.2. *Let D be a dataset, and for each similarity threshold s_i, construct a similarity graph $G_i(V, E_i)$, where V represents the data points and $(u, v) \in E_i$ if and only if the cosine similarity between u and v exceeds s_i. Let $H_i \subseteq V$ be the set of k samples selected by the max coverage algorithm on G_i, and let c_i denote the coverage achieved by H_i. For any two thresholds s_i and s_j such that $s_j < s_i$, the similarity graph $G_j(V, E_j)$ has a coverage $c_j \geq c_i$ when maximally covered by k samples.*

Proof. Consider two similarity thresholds s_i and s_j such that $s_j < s_i$. The corresponding similarity graphs $G_i(V, E_i)$ and $G_j(V, E_j)$ are constructed by adding edges between data points whose cosine similarity exceeds s_i and s_j, respectively. Since $s_j < s_i$, it follows that $E_i \subseteq E_j$; that is, G_j includes all the edges from G_i, possibly with additional edges.

Now, let $H_i \subseteq V$ be the set of k samples selected by the max coverage algorithm on G_i, which achieves coverage c_i. The coverage c_i is defined as the proportion of vertices in V that are adjacent to at least one vertex in H_i. Since $E_i \subseteq E_j$, the set of neighbors of each vertex in H_i in G_i is a subset of the neighbors of the same vertex in G_j. Therefore, the coverage achieved by H_i in G_j is at least as large as the coverage in G_i. More formally, if H_j is the set of k samples selected by the

max coverage algorithm on G_j, we have:

$$c_j = \left| \bigcup_{v \in H_j} N_j(v) \right| \quad \text{and} \quad c_i = \left| \bigcup_{v \in H_i} N_i(v) \right|,$$

where $N_j(v)$ and $N_i(v)$ denote the neighborhoods of v in G_j and G_i, respectively. Since $E_i \subseteq E_j$, we have $N_i(v) \subseteq N_j(v)$ for all $v \in V$, implying that the coverage in G_j is at least as large as the coverage in G_i. Therefore, $c_j \geq c_i$. □

This theorem guarantees that as we lower the threshold (adding edges), coverage can only increase, enabling efficient binary search for the highest threshold achieving target coverage.

We crucially note that the max-coverage problem is NP-hard (Feige, 1998), and in practice we use the greedy approximation algorithm (Hochbaum, 1996) which provides a $(1 - 1/e)$ approximation guarantee for submodular maximization. While Theorem 2.2 applies to exact solutions, we empirically validate in Section 3 that monotonicity persists under greedy approximation across all tested datasets, justifying the binary search procedure.

Step 4: Greedy Maximum Coverage. Given the optimally thresholded graph, we apply the greedy max-cover algorithm: iteratively select the node covering the most uncovered samples (i.e. highest degree among remaining nodes), mark that node and its neighbors as covered, and repeat until k nodes are selected.

For very large datasets, computing the optimal threshold even via binary search on the full graph may be expensive. In Appendix D, we prove that the threshold can be accurately estimated on a small random subsample of the data with bounded error via a Hoeffding-type concentration inequality. Empirically, we show that subsampling as little as 20% of the data yields thresholds that generalize to the full dataset with negligible coverage deviation. This two-stage approach—tune threshold on a subsample, apply to the full data—enables ACS to scale to arbitrarily large corpora.

Summary of ACS flexibility: The framework accommodates various design choices: (1) any embedding method capturing task-relevant semantics, (2) approximate graph construction via LSH or partitioning Dasgupta et al. (2011), (3) threshold estimation on subsamples for scalability, and (4) alternative coverage targets depending on application needs. This generality makes ACS broadly applicable across domains and scales while maintaining theoretical guarantees on the core coverage objective.

2.3 Comparative Experiments

We evaluate all methods by fine-tuning a BERT model (Devlin, 2018) on the selected k samples and measuring F1-score on held-out test sets. We employ $\text{BERT}_{\text{base}}$ uncased (108M parameters), fine-tuning for task-specific epochs (further detailed in Section 3). Pre-trained weights initialize the encoder, while the final classification layer (2048 units) is randomly initialized from $\mathcal{N}(0, 0.02^2)$ following standard practice (Devlin, 2018; Dodge et al., 2020). Training uses batch size 16, learning rate of 2×10^{-5}, and dropout 0.1. All experiments average over 5 random seeds on GPUs with 16GB RAM.

Beyond the standard classification results, we compute SelfBLEU (Zhu et al., 2018), which measures lexical similarity between sentences within a dataset. Higher SelfBLEU indicates higher self-similarity (redundancy), so we report inverse SelfBLEU as a diversity measure. This provides an independent validation that ACS-selected subsets are indeed more diverse than those from baseline methods.

Why BERT? We deliberately use BERT rather than more recent LLMs to isolate the data selection variable from model capability. Modern LLMs possess extensive world knowledge that could mask differences in training data quality. BERT serves as a sensitive probe: if ACS enables BERT to match full-dataset performance with only 20% of samples, this demonstrates that our method identifies the core semantic subsets. Moreover, ACS operates on embedding geometry and is agnostic to the downstream model—improvements transfer to any architecture.

3 EMPIRICAL ANALYSIS OF ACS

Before evaluating ACS on downstream tasks, we validate two critical assumptions underlying our method. First, we verify that the monotonicty property established in Theorem 2.2 for *exact* max coverage holds empirically when using the greedy approximation algorithm, This validates our binary search procedure for threshold selection. Second, we investigate the optimal coverage level, demonstrating that moderate coverage ($c < 1.0$) consistently outperforms full coverage. This is a non-trivial finding that reveals the importance of strategic sample exclusion.

3.1 EMPIRICAL VALIDATION OF MONOTONICITY

Theorem 2.2 proves that coverage increases monotonically as the similarity threshold decreases, but this guarantee applies only to the exact (NP-hard) max coverage solution. In practice, we use the greedy approximation (Hochbaum, 1996), which provides a $(1 - 1/e)$ approximate guarantee but does not theoretically preserve monotonicity.

Experimental Design. We empirically test whether greedy max coverage exhibits monotonic behavior across varying similarity thresholds. Using the synthetic SST2 sentiment analysis dataset (Socher et al., 2013) (6,000 movie reviews labeled as positive/negative), we embed samples with Gecko embeddings (Lee et al., 2024) and construct similarity graphs at multiple fixed thresholds: $\tau \in [0, 1]$. For each threshold, we apply greedy max coverage to select subsets of varying sizes ($k = 10, 20, ..., 100$) and measure the resulting coverage.

Results. Figure 1 (left) shows coverage as a function of k for each fixed threshold. Each curve corresponds to a different threshold value, with lower thresholds (denser graphs, more edges) shown in darker colors. The results unambiguously demonstrate monotonic behavior: as the threshold decreases, coverage curves either remain identical or shift upward, never downward. This confirms that lower thresholds (adding edges) enable equal or higher coverage, exactly as predicted by theory, despite using greedy approximation.

We crucially note that: (1) all curves achieve minimum coverage of k/N (i.e. selecting k samples covers at least themselves, (2) lower thresholds reach full coverage ($c = 1.0$) more rapidly, as expected from denser graph connectivity, (3) at very low thresholds, coverage saturates quickly, i.e. the graph becomes highly connected which allows a small number of nodes to cover most of the dataset, (4) monotonicity holds across all tested values of k, not just in aggregate.

We replicate this experiment on FewRel (relation extraction) and CrossNER (named entity recognition) datasets, observing consistent monotonicity in all cases (see Appendix B.1). This empirical validation across diverse tasks and dataset sizes strongly supports the practical validity of our binary search procedure, even when using the computationally efficient greedy approximation.

3.2 DETERMINING THE OPTIMAL COVERAGE LEVEL

Intuitive Expectation vs. Empirical Reality. One might assume that full coverage ($c = 1.0$), which ensures every sample is either selected or adjacent to a selected sample, would maximize downstream model performance by preserving complete representational breadth. However, this intuition neglects two factors: (1) the least-connected samples (those only reached as c approaches 1.0) may represent outliers, low-quality generations, or truly redundant variants; (2) training on such samples could harm generalization by introducing noise or over-fitting to generation artifacts.

Experimental Design. To identify the optimal coverage level, we conduct a systematic sweep over target coverage values $c \in (0.1, 1]$ for the synthetic SST2 dataset. For each coverage target, we fix the subset size at k = 300 samples (5% of the 6,000-sample corpus) and use ACS to select a subset achieving that target coverage via binary search on the threshold. We then fine-tune $\text{BERT}_{\text{base}}$ models on each selected subset and evaluate F1-score on the human-annotated SST2 test set.

Results. Figure 1 (right) plots downstream F1-score as a function of target coverage for different values of $k \in \{100, 200, 300, 400, 500, 600\}$. Several clear patterns emerge. First, performance increases sharply as coverage rises from low values, peaks at moderate coverage ($c \approx 0.8 - 0.9$), and

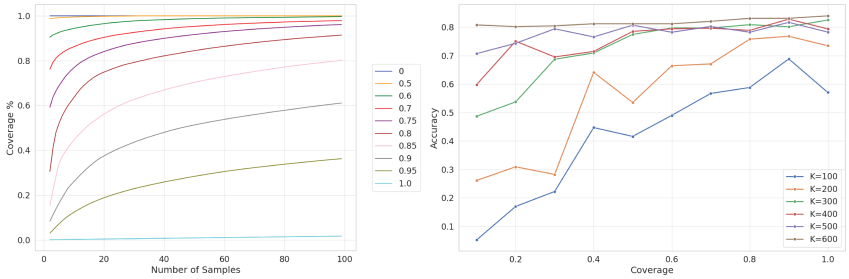

Figure 1: (L) Coverage of data increases with k or when decreasing the similarity threshold. Colors correspond to the fixed similarity thresholds depicted in the legend. (R) Model accuracy as a function of coverage level for the sentiment analysis tasks. Performance peaks at a coverage level below 1.0.

then declines or plateaus as coverage approaches 1.0. This trend holds consistently across all tested values of k. The peak performance occurs at $c \approx 0.9$ for most values of k, meaning that deliberately excluding the least-connected 10% of samples improves model accuracy compared to full coverage. This exclusion effect is most pronounced for smaller k (aggressive pruning), where selecting truly representative samples is most critical. The performance drop from $c = 0.9$ to $c = 1.0$ is modest but consistent, suggesting that the marginal samples reached at full coverage add noise without informational value. Lastly, we note that for $c < 0.5$, performance is substantially degraded, confirming that representational breadth matters, and overly aggressive pruning loses important semantic regions.

Interpretation. These results reveal that strategic exclusion improves generalization. The least-connected samples in the similarity graph—those requiring very low thresholds to connect—likely represent: (a) near-duplicates of already-selected samples (redundancy), (b) outliers or low-quality LLM generations (noise), or (c) overly specific edge cases with limited generalization value. By targeting $c \approx 0.9$, ACS focuses on the core semantic structure while filtering the problematic tails of the distribution.

Practical implications. This finding justifies ACS's use of moderate coverage (default $c = 0.9$) as a form of implicit quality filtering, without requiring explicit quality scoring or LLM-based rating. The coverage parameter provides an interpretable, theoretically grounded control over the diversity-redundancy tradeoff, adapting automatically to the dataset's similarity structure via binary search.

We replicate this coverage sweep on FewRel and CrossNER, observing qualitatively similar curves with optimal coverage in the range $c \in [0.8, 0.95]$ (Appendix B.1). The consistency across tasks suggests that moderate coverage is a robust default, though task-specific tuning may yield marginal further improvements.

4 Fine-Tuning for Downstream Tasks

We now evaluate ACS against established baseline methods across three diverse NLP benchmarks, demonstrating consistent advantages in both model performance and subset diversity. Our experiments span sequence-level tasks (sentiment analysis with 2 classes, relation extraction with 64 classes) and token-level classification (named entity recognition with 14 entity types), validating ACS's effectiveness across varying task complexities and label space sizes.

A key finding emerges across all tasks: ACS achieves performance equivalent to training on full synthetic datasets using only 10–30% of the data. This dramatic reduction in training data translates directly to proportional savings in computational cost, training time, and energy consumption. Moreover, we observe a strong correlation between subset diversity (measured by inverse SelfBLEU) and downstream performance: ACS-selected subsets consistently exhibit higher diversity than all baselines, explaining their superior generalization.

We here note that while we evaluate performance across a range of subset sizes k to thoroughly characterize behavior, in practice the optimal k can be determined efficiently by monitoring validation set performance, stopping when further data addition provides diminishing returns.

4.1 SEQUENCE-LEVEL TASK: SENTIMENT ANALYSIS

Task and Dataset. We evaluate binary sentiment classification using the synthetic SST2 corpus from (Ding et al., 2022), comprising $N = 6,000$ movie reviews balanced between positive and negative labels. Following prior work (Ding et al., 2022), we fine-tune $\text{BERT}_{\text{base}}$ for 32 epochs with early stopping on subsets selected by each method.

Results: The "Less is More" Effect. Figure 2 (left column, top row) presents F1-scores averaged over five random initializations for the SST2 synthetic dataset. ACS demonstrates substantial advantages, particularly at aggressive pruning rates where intelligent selection matters most. Most strikingly, ACS matches full-dataset performance (horizontal dashed line at F1 = 0.8176) using only 10% of the synthetic data (600 samples), while baselines require 40–60% to reach comparable accuracy. At 20% data retention, ACS achieves F1 = 0.8368, actually exceeding full-dataset performance by +0.0192, a result consistent with our finding in Section 3 that strategic exclusion of redundant samples improves generalization.

ACS outperforms all methods across nearly all subset sizes (winning 9 of 10 comparisons), with advantages most pronounced at $k \leq 0.3 \cdot N$. Random sampling performs surprisingly well for this relatively simple binary task, but still lags ACS at low data regimes. Training-dynamics methods (EL2N, forgetting) show inconsistent performance, with EL2N notably underperforming at several intermediate sizes, likely because these methods prioritize "hard" examples that may be mislabeled in synthetic data. Prototypicality and AlpaGasus provide stronger competition but still trail ACS at critical low-data regimes.

Why ACS Wins: Diversity Drives Generalization. The lower row of Figure 2 reveals the mechanism behind ACS's advantage. ACS-selected subsets exhibit consistently lower SelfBLEU scores (higher diversity) across all k values. This enhanced diversity ensures exposure to varied linguistic patterns, sentiment expressions, and edge cases which enables better generalization despite reduced training data. Baselines, particularly prototypicality, tend toward redundant class-representative samples rather than diverse coverage.

We note that even random sampling achieves reasonable performance after aggressive pruning (though still inferior to ACS), suggesting that for binary sentiment classification (a relatively well-defined task with clear linguistic signals) a moderate number of diverse examples suffices for effective classifier training. This makes ACS's advantages even more impressive: if the task is already amenable to data reduction, ACS's principled selection identifies the minimal sufficient subset.

4.2 SEQUENCE-LEVEL TASK: RELATION EXTRACTION

Task and Dataset. Relation extraction presents a substantially harder challenge than sentiment analysis due to its 64-way classification over relation types. Using the synthetic FewRel corpus (Han et al., 2018; Ding et al., 2022) with $N = 12,800$ examples uniformly distributed across relations, the task requires predicting the labeled semantic relation between two marked entities in a sentence (e.g., "head of government" for "Chester Alan Arthur, 21st President..."). This increased label space demands both greater diversity and semantic precision. We fine-tune BERTbase for 3 epochs following prior work.

Results: Consistent Advantages at Scale. Figure 2 (middle column) demonstrates that ACS's advantages persist and strengthen for this more complex task. ACS matches full-dataset performance (F1 = 0.3729) using approximately 30% of the data (3,800 samples), while baselines require 50–70% to reach comparable accuracy. At 30% retention, ACS achieves F1 = 0.3732, slightly exceeding full-dataset training and again confirming that quality trumps quantity.

The performance gaps between ACS and baselines widen compared to sentiment analysis. At 10% data retention, ACS achieves F1 = 0.2642 versus random's 0.2293 (+0.0349, a 15% relative improvement). At 20%, the gap remains substantial: ACS scores 0.3352 versus the next-best EL2N at 0.3191 (+0.0161). This pattern suggests that ACS's advantages compound as task complexity increases. When label spaces are large and semantic distinctions are subtle, intelligent coverage-based selection becomes essential.

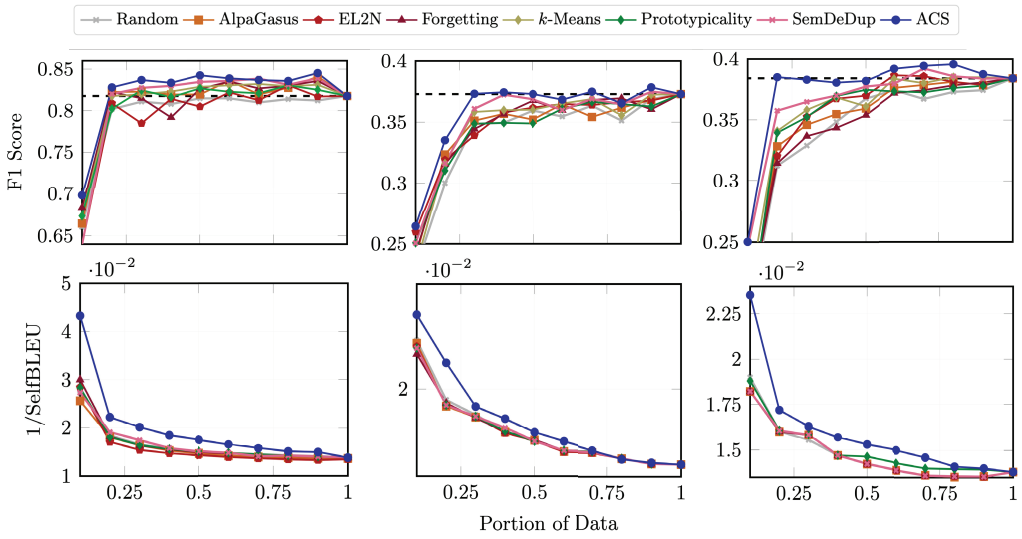

Figure 2: F1 scores (top row) and SelfBLEU diversity (bottom row) for the SST2 (L), FewRel (C) and CrossNER (R) datasets as a function of downsampled subset size, comparing downsampling methods. Horizontal dotted line on top row represents model performance when trained on all available data. Data is depicted in tabular form in Appendix C.

Diversity Correlates with Performance. The diversity panel shows ACS consistently produces the most diverse subsets (lowest SelfBLEU), with gaps to baselines widening as k decreases. For relation extraction, this diversity is particularly valuable: the task requires exposure to varied entity types, syntactic constructions, and contextual patterns to distinguish between 64 fine-grained relations. Prototypicality and AlpaGasus, while selecting high-quality individual samples, fail to ensure comprehensive coverage of this semantic space.

Practical Significance. Reducing training data from 12,800 to 3,800 samples (70% reduction) while maintaining performance represents substantial computational savings. For organizations fine-tuning relation extraction models across multiple domains or frequently updating on new synthetic data, this efficiency gain scales linearly with the number of training runs.

4.3 Token-Level Task: Named Entity Recognition

Task and Dataset. Named entity recognition (NER) requires labeling each token with one of 14 entity classes (or null) in the AI domain using the synthetic CrossNER corpus (Liu et al., 2021) with $N = 3,000$ sentences. For example, in "We evaluated BERT using the SQuAD benchmark...", the model must identify BERT (Tool), SQuAD (Dataset), etc. Importantly, while NER is a token-level task, we apply ACS at the sentence level, embedding entire sentences and selecting those that maximize coverage. This design choice treats sentences as the unit of diversity, ensuring selected examples expose the model to varied entity co-occurrence patterns and contexts. We fine-tune a NER-specific BERTbase model (Rajapakse et al., 2024) for 50 epochs with early stopping.

Results: ACS Excels on Fine-Grained Tasks. Figure 2 (right column) shows ACS's most dramatic performance advantages. ACS matches full-dataset performance (F1 = 0.3842) using only 20% of the data (600 sentences), while baselines require 60–80%. At 10% retention, ACS achieves F1 = 0.2502 versus prototypicality's 0.1917 (+0.0585, a 31% relative improvement), the largest gap observed across all experiments. At 20%, ACS scores 0.3851, actually exceeding full-dataset training by +0.0009, while the next-best prototypicality reaches only 0.3394.

We speculate that token-level tasks are particularly sensitive to training data diversity because effective NER requires exposure to: (1) varied entity types in different contexts, (2) diverse sentence structures and lengths, (3) different entity co-occurrence patterns. Random or prototypicality-based

selection risks oversampling common sentence patterns while missing rare but important entity configurations. ACS's coverage objective ensures representation across the full space of entity-context combinations, explaining its dramatic advantages.

Diversity Gaps Widen. The diversity panel shows ACS producing substantially more diverse subsets than any baseline, with gaps increasing as k decreases. This is expected: as we aggressively prune data, the difference between intelligent coverage-based selection and heuristic methods becomes most apparent. Baselines converge toward similar, redundant samples; ACS maintains broad coverage.

5 Discussion

Our experiments establish a consistent pattern: carefully curated synthetic subsets systematically outperform training on full datasets. Across sentiment analysis, relation extraction, and named entity recognition, ACS achieves equivalent or superior performance using only 10–30% of synthetic data, with advantages most pronounced for complex tasks where comprehensive coverage is critical.

In addition to identifying subsets comprising 10–30% of original data that match full-dataset performance, ACS operates on embedding geometry without fitting models. Selection completes before downstream training begins, avoiding the computational overhead that synthetic data curation aims to eliminate. Moreover, ACS uses only semantic similarity, making it applicable to unsupervised scenarios, noisy labels, or pretraining data curation

As such, ACS provides a principled approach to synthetic data curation through graph-based maximum coverage with adaptive threshold tuning. By achieving equivalent performance with fractional amounts of data across diverse tasks, ACS demonstrates that quality, as measured by diversity and coverage, systematically outperforms quantity for LLM-generated corpora. As synthetic data becomes central to ML pipelines, principled selection methods like ACS are essential for realizing the efficiency potential of LLMs as data generators while avoiding computational waste from redundancy.

References

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

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Andreas Bunte, Frank Richter, and Rosanna Diovisalvi. Why it is hard to find ai in smes: A survey from the practice and how to promote it. In *ICAART (2)*, pp. 614–620, 2021.

CJ Carey, Jonathan Halcrow, Rajesh Jayaram, Vahab Mirrokni, Warren Schudy, and Peilin Zhong. Stars: Tera-scale graph building for clustering and learning. *Advances in Neural Information Processing Systems*, 35:21470–21481, 2022.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, et al. Alpagasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701*, 2023.

Xiusi Chen, Jyun-Yu Jiang, and Wei Wang. Scalable graph representation learning via locality-sensitive hashing. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 3878–3882, 2022.

Anirban Dasgupta, Ravi Kumar, and Tamás Sarlós. Fast locality-sensitive hashing. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1073–1081, 2011.

Shumin Deng, Ningyu Zhang, Zhanlin Sun, Jiaoyan Chen, and Huajun Chen. When low resource nlp meets unsupervised language model: Meta-pretraining then meta-learning for few-shot text classification (student abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 13773–13774, 2020.

Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Bosheng Ding, Linlin Liu, Lidong Bing, Canasai Kruengkrai, Thien Hai Nguyen, Shafiq Joty, Luo Si, and Chunyan Miao. Daga: Data augmentation with a generation approach for low-resource tagging tasks. *arXiv preprint arXiv:2011.01549*, 2020.

Bosheng Ding, Chengwei Qin, Linlin Liu, Yew Ken Chia, Shafiq Joty, Boyang Li, and Lidong Bing. Is gpt-3 a good data annotator? *arXiv preprint arXiv:2212.10450*, 2022.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*, 2020.

Alessandro Epasto, Andrés Muñoz Medina, Steven Avery, Yijian Bai, Robert Busa-Fekete, CJ Carey, Ya Gao, David Guthrie, Subham Ghosh, James Ioannidis, et al. Clustering for private interest-based advertising. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 2802–2810, 2021.

Uriel Feige. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4): 634–652, 1998.

Saumya Gandhi, Ritu Gala, Vijay Viswanathan, Tongshuang Wu, and Graham Neubig. Better synthetic data by retrieving and transforming existing datasets. *arXiv preprint arXiv:2404.14361*, 2024.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120, 2023.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 513–520, 2011.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International journal of computer vision*, 129(6):1789–1819, 2021.

Chengcheng Guo, Bo Zhao, and Yanbing Bai. Deepcore: A comprehensive library for coreset selection in deep learning. In *International Conference on Database and Expert Systems Applications*, pp. 181–195. Springer, 2022.

Jonathan Halcrow, Alexandru Mosoi, Sam Ruth, and Bryan Perozzi. Grale: Designing networks for graph learning. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2523–2532, 2020.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. *arXiv preprint arXiv:1810.10147*, 2018.

Shuang Hao, Wenfeng Han, Tao Jiang, Yiping Li, Haonan Wu, Chunlin Zhong, Zhangjun Zhou, and He Tang. Synthetic data in ai: Challenges, applications, and ethical implications. *arXiv preprint arXiv:2401.01629*, 2024.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3309–3326, 2022.

Ruidan He, Linlin Liu, Hai Ye, Qingyu Tan, Bosheng Ding, Liying Cheng, Jia-Wei Low, Lidong Bing, and Luo Si. On the effectiveness of adapter-based tuning for pretrained language model adaptation. *arXiv preprint arXiv:2106.03164*, 2021.

Dorit S Hochbaum. Approximating covering and packing problems: set cover, vertex cover, independent set, and related problems. In *Approximation algorithms for NP-hard problems*, pp. 94–143. 1996.

Tom Hosking, Phil Blunsom, and Max Bartolo. Human feedback is not gold standard. In *The Twelfth International Conference on Learning Representations*.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. PMLR, 2017.

Hsun-Yu Kuo, Yin-Hsiang Liao, Yu-Chieh Chao, Wei-Yun Ma, and Pu-Jen Cheng. Not all llm-generated data are equal: Rethinking data weighting in text classification. *arXiv preprint arXiv:2410.21526*, 2024.

Alexey Kurakin, Natalia Ponomareva, Umar Syed, Liam MacDermed, and Andreas Terzis. Harnessing large-language models to generate private synthetic text. *arXiv preprint arXiv:2306.01684*, 2023.

Jinhyuk Lee, Zhuyun Dai, Xiaoqi Ren, Blair Chen, Daniel Cer, Jeremy R Cole, Kai Hui, Michael Boratko, Rajvi Kapadia, Wen Ding, et al. Gecko: Versatile text embeddings distilled from large language models. *arXiv preprint arXiv:2403.20327*, 2024.

Yinheng Li, Rogerio Bonatti, Sara Abdali, Justin Wagle, and Kazuhito Koishida. Data generation using large language models for text classification: An empirical case study. *arXiv preprint arXiv:2407.12813*, 2024.

Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. Synthetic data generation with large language models for text classification: Potential and limitations. *arXiv preprint arXiv:2310.07849*, 2023.

Hui Lin, Jeff Bilmes, and Shasha Xie. Graph-based submodular selection for extractive summarization. In *2009 IEEE Workshop on Automatic Speech Recognition & Understanding*, pp. 381–386. IEEE, 2009.

Ruibo Liu, Jerry Wei, Fangyu Liu, Chenglei Si, Yanzhe Zhang, Jinmeng Rao, Steven Zheng, Daiyi Peng, Diyi Yang, Denny Zhou, et al. Best practices and lessons learned on synthetic data. In *First Conference on Language Modeling*, 2024.

Zihan Liu, Yan Xu, Tiezheng Yu, Wenliang Dai, Ziwei Ji, Samuel Cahyawijaya, Andrea Madotto, and Pascale Fung. Crossner: Evaluating cross-domain named entity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 13452–13460, 2021.

Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. On llms-driven synthetic data generation, curation, and evaluation: A survey. *arXiv preprint arXiv:2406.15126*, 2024.

Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. Generating training data with language models: Towards zero-shot language understanding. *Advances in Neural Information Processing Systems*, 35:462–477, 2022.

Takeru Miyato, Andrew M Dai, and Ian Goodfellow. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*, 2016.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. *Advances in neural information processing systems*, 34:20596–20607, 2021.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Thilina C Rajapakse, Andrew Yates, and Maarten de Rijke. Simple transformers: Open-source for all. In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, pp. 209–215, 2024.

Gaurav Sahu, Pau Rodriguez, Issam Laradji, Parmida Atighehchian, David Vazquez, and Dzmitry Bahdanau. Data augmentation for intent classification with off-the-shelf large language models. In *Proceedings of the 4th Workshop on NLP for Conversational AI*, pp. 47–57, 2022.

Sarath Shekkizhar, Neslihan Bulut, Mohamed Farghal, Sasan Tavakkol, MohammadHossein Bateni, and Animesh Nandi. Data sampling using locality sensitive hashing for large scale graph learning. 2023.

Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, et al. Beyond human data: Scaling self-training for problem-solving with language models. *Transactions on Machine Learning Research*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35:19523–19536, 2022.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9275–9293, 2020.

Ruixiang Tang, Xiaotian Han, Xiaoqian Jiang, and Xia Hu. Does synthetic data generation of llms help clinical text mining? *arXiv preprint arXiv:2303.04360*, 2023.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*, 2022.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*, 2018.

A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*, 2019.

Xiaobo Xia, Jiale Liu, Jun Yu, Xu Shen, Bo Han, and Tongliang Liu. Moderate coreset: A universal method of data selection for real-world data-efficient deep learning. In *The Eleventh International Conference on Learning Representations*, 2022.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. Zerogen: Efficient zero-shot learning via dataset generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11653–11669, 2022.

Haizhong Zheng, Rui Liu, Fan Lai, and Atul Prakash. Coverage-centric coreset selection for high pruning rates. In *The Eleventh International Conference on Learning Representations*.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pp. 1097–1100, 2018.

A Related Work

Large Language Models. LLMs, built upon the transformer architecture introduced by (Vaswani, 2017), have transformed language processing, achieving unprecedented performance across a broad spectrum of tasks including language modeling, translation, classification, and question-answering (Brown, 2020; Rae et al., 2021; Taylor et al., 2022). These models leverage massive-scale pretraining on extensive datasets to encode rich linguistic and factual knowledge, enabling fluent and contextually relevant text generation (Team et al., 2024). Consequently, the sophistication of LLM-generated content increasingly blurs the line between synthetic and authentic human-written text (Hartvigsen et al., 2022; Sahu et al., 2022; Tang et al., 2023; Ye et al., 2022).

Synthetic Training Data Generation. High-quality datasets crucially underpin the performance and generalization capabilities of modern machine learning systems. However, acquiring diverse and representative labeled data from human annotators is frequently costly, labor intensive, and fraught with privacy or ethical challenges (Kurakin et al., 2023; Gilardi et al., 2023; Hosking et al.; Singh et al.). Moreover, human-generated annotations inherently carry biases or inconsistencies, potentially limiting their effectiveness in certain contexts. To overcome these limitations, synthetic data generation has emerged as a promising alternative, aimed at artificially populating underrepresented data regions and mitigating biases or gaps in existing datasets (Gandhi et al., 2024; Liu et al., 2024; Li et al., 2023).

To address data scarcity in specialized or emerging domains, researchers frequently employ data augmentation techniques to enhance model robustness and accuracy (Ding et al., 2020; Wei & Zou, 2019). Moreover, semi-supervised learning (Miyato et al., 2016), multi-task learning (Glorot et al., 2011), unsupervised pretraining (Devlin, 2018; Raffel et al., 2020), and few-shot learning (Deng et al., 2020; He et al., 2021) constitute alternative frameworks for learning from limited labeled examples. However, while effective in certain contexts, these approaches typically presume access to at least some high-quality human-generated examples as seed data, limiting their broader applicability.

Leveraging LLMs for Synthetic Data. LLMs offer a compelling approach to synthetic data generation due to their fluency, versatility, and capacity to mimic diverse linguistic styles and content structures (Ding et al., 2022). Recent studies have demonstrated promising outcomes leveraging prompt-based methods for generating training data for NLP tasks (Long et al., 2024). The effectiveness of synthetic datasets produced by these models depends critically on task characteristics, including the complexity of label spaces (Ding et al., 2022), the inherent subjectivity or ambiguity of the task (Li et al., 2023), and crucially, the diversity and representativeness of generated samples (Hao et al., 2024). Though the models are promising, these factors can impede naively employed models trained on synthetic datasets, potentially exacerbating redundancy and bias. Thus, underscoring the necessity of methods to carefully select or filter synthetic samples to maximize utility and minimize detrimental impacts.

Data Filtering and Downsampling. Filtering datasets to identify informative subsets for training constitutes a widely explored solution to the challenges posed by redundancy and imbalance. Conventional data selection techniques can be broadly categorized into three approaches: training-dynamics-based methods, influence-based scoring, and similarity-based selection.

Training dynamics methods leverage how models interact with data during learning. Dataset cartography (Swayamdipta et al., 2020) identifies difficult or ambiguous samples through repeated training runs, emphasizing points near decision boundaries. EL2N scoring (Paul et al., 2021) ranks samples by the L2 distance between predictions and true labels across early training checkpoints, prioritizing persistently challenging examples. Forgetting scores (Toneva et al., 2018) count prediction transitions between correct and incorrect classifications, targeting samples that prove difficult to memorize. While effective, these methods require multiple training iterations to compute scores, incurring substantial computational overhead, precisely what synthetic data selection aims to reduce.

Influence-based methods quantify individual sample contributions by approximating how their removal would alter model parameters (Koh & Liang, 2017) or by measuring gradient similarity (Guo et al., 2022). Prototypicality assessments (Sorscher et al., 2022) compute class-specific embeddings and prioritize samples closest to class centroids. However, these approaches often select represen-

tative but potentially redundant samples, as proximity to centroids favors common patterns over diverse edge cases.

Similarity-based and coverage methods construct graphs or clusters over data representations to ensure diversity. Recent coreset selection work (Zheng et al.; Xia et al., 2022) applies coverage objectives to general dataset pruning, while deduplication methods (Abbas et al., 2023) remove near-duplicate samples in pretraining corpora. However, these approaches typically use fixed similarity thresholds or require dataset-specific tuning, lacking adaptive mechanisms to balance coverage and sparsity. Graph-based submodular maximization for summarization (Lin et al., 2009) shares conceptual similarities but targets document-level diversity rather than training data representativeness with semantic constraints.

LLM-based filtering has emerged as an alternative paradigm. AlpaGasus (Chen et al., 2023) employs GPT-3.5 to assign quality ratings to synthetic input-output pairs, retaining only highly-rated samples. While this improves over random selection, it requires repeated expensive LLM queries (effectively training another model to rate the first model's outputs), produces black-box scores lacking interpretability, and necessitates careful threshold tuning to determine the quality cutoff. The approach also inherits biases from the rating model, potentially amplifying systematic preferences in the generator LLM (Li et al., 2024; Kuo et al., 2024).

ACS distinguishes itself through several key properties: (1) selection operates purely on embedding geometry, avoiding iterative model fitting, (2) binary search systematically identifies optimal graph sparsity for target coverage, eliminating manual tuning, (3) degree constraints prevent selection of redundant hub samples, emphasizing local representativeness over global centrality, (4) monotonicity guarantees justify the search procedure, while Hoeffding bounds enable scalable approximation, (5) ACS consistently achieves comparable performance with 10-30% of data, substantially outperforming prior methods at aggressive pruning rates where intelligent selection matters most. By formulating synthetic data selection as maximum coverage with adaptive graph sparsification, ACS provides a principled, efficient, and interpretable solution that establishes quality and diversity, not quantity, as the drivers of synthetic data utility.

B Omitted Empirical Results

We here present the empirical analysis of Section 3 on the FewRel and CrossNER datasets. We further present a sensitivity analysis to the max degree parameter for all of the datasets.

B.1 Empirical Analysis of ACS

We begin with the empirical ACS validation of Section 3 for the remaining datasests. In both instances, we observe consistent monotonicity in the coverage as a function of k-selection with decreasing similarity thresholds, as well as improved downstream task performance with coverage values less than 1.0. Figure 3 presents the empirical results for the FewRel dataset and Figure 4 for CrossNER. In both instances, the greedy approximation to max coverage exhibits monotonicity as needed for the binary search procedure. We further see that full coverage is non-optimal in most instances, further motivating our usages of coverage = 0.9 throughout the experimental results.

C Alternative Data Presentation

We here present the results of Section 4 in tabular form for maximal clarity.

D Scalability of Adaptive Coverage Sampling

In large-scale settings, the computational cost of optimizing the similarity threshold τ for ACS can become prohibitive due to the $O(n^2)$ complexity of evaluating pairwise similarities. Though we can speed up such computations with methods such as Locality Sensitive Hasing (LHS) or hop-spanner methods (Carey et al., 2022; Epasto et al., 2021; Halcrow et al., 2020), we further propose a scalable variant of ACS that conducts threshold selection on a small random subset of the data. For a desired downsampling value of $k \ll N$, we uniformly at random select a small subgraph of $N' < N$ nodes

%	Random	AlpaGasus	Forgetting	Prototypicality	EL2N	k-Means	SemDeDup	ACS
10	0.6605	0.6647	0.6830	0.6737	0.6387	0.6603	0.6701	**0.6982**
20	0.8108	0.8227	0.8149	0.8246	0.7848	0.8175	0.8201	**0.8368**
30	0.8079	0.8195	0.7919	0.8163	0.8142	0.8198	0.8274	**0.8335**
40	0.8238	0.8223	0.8168	**0.8319**	0.7737	0.8230	0.8298	0.8312
50	0.8148	0.8195	0.8262	0.8267	0.8049	0.8287	0.8346	**0.8425**
60	0.8148	0.8360	0.8350	0.8227	0.8220	0.8304	0.8358	**0.8388**
70	0.8099	0.8195	0.8268	0.8209	0.8125	0.8321	**0.8385**	0.8368
80	0.8139	0.8275	0.8298	0.8304	0.8305	0.8294	0.8311	**0.8357**
90	0.8122	0.8409	0.8357	0.8251	0.8170	0.8311	0.8390	**0.8452**
100	0.8176	0.8176	0.8176	0.8176	0.8176	0.8176	0.8176	0.8176

Table 1: Tabulated F1 results for SST2 dataset with added baselines of k-Means and DeDup .

%	Random	AlpaGasus	Forgetting	Prototypicality	EL2N	k-Means	SemDeDup	ACS
10	0.2293	0.2393	0.2427	0.2508	0.2597	0.2314	0.2506	**0.2642**
20	0.3000	0.3235	0.3157	0.3102	0.3191	0.3164	0.3162	**0.3352**
30	0.3486	0.3512	0.3444	0.3487	0.3391	0.3583	0.3609	**0.3732**
40	0.3494	0.3568	0.3566	0.3494	0.3579	0.3598	0.3724	**0.3744**
50	0.3595	0.3520	0.3677	0.3489	0.3619	0.3601	0.3687	**0.3730**
60	0.3546	0.3647	0.3597	0.3609	0.3648	0.3652	0.3598	**0.3684**
70	0.3634	0.3542	0.3682	0.3665	0.3640	0.3689	0.3690	**0.3749**
80	0.3512	0.3618	0.3695	0.3647	**0.3663**	0.3555	0.3644	0.3654
90	0.3689	0.3681	0.3604	0.3628	0.3676	0.3724	0.3749	**0.3785**
100	0.3729	0.3729	0.3729	0.3729	0.3729	0.3729	0.3729	0.3729

Table 2: Tabulated F1 results for FewRel dataset with added baselines of k-Means and SemDeDup .

%	Random	AlpaGasus	Forgetting	Prototypicality	EL2N	k-Means	SemDeDup	ACS
10	0.1820	0.1850	0.1905	0.1789	0.1917	0.2031	0.2456	**0.2502**
20	0.3122	0.3283	0.3142	0.3394	0.3202	0.3409	0.3575	**0.3851**
30	0.3289	0.3459	0.3366	0.3523	0.3526	0.3582	0.3648	**0.3830**
40	0.3483	0.3547	0.3434	0.3696	0.3674	0.3684	0.3700	**0.3806**
50	0.3675	0.3593	0.3538	0.3747	0.3699	0.3611	0.3774	**0.3820**
60	0.3745	0.3764	0.3721	0.3736	0.3867	0.3839	0.3798	**0.3921**
70	0.3673	0.3788	0.3745	0.3727	0.3860	0.3803	0.3923	**0.3945**
80	0.3731	0.3814	0.3785	0.3763	0.3812	0.3852	0.3859	**0.3958**
90	0.3745	0.3787	0.3809	0.3781	0.3788	0.3851	0.3840	**0.3877**
100	0.3842	0.3842	0.3842	0.3842	0.3842	0.3842	0.3842	0.3842

Table 3: Tabulated F1 results for CrossNER dataset with added baselines of k-Means and DeDup .

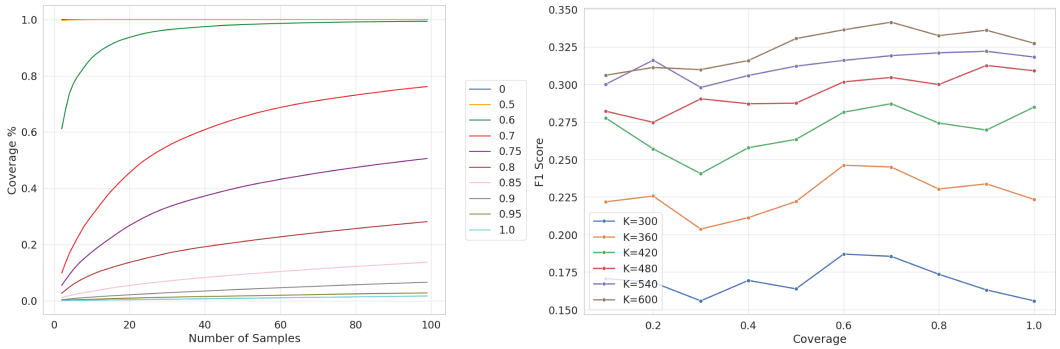

Figure 3: Empirical results for the FewRel dataset. (L) Coverage of data increases with k or when decreasing the similarity threshold. Colors correspond to the fixed similarity thresholds depicted in the legend. (R) Model accuracy as a function of coverage level for the sentiment analysis tasks. Performance peaks at a coverage level below 1.0.

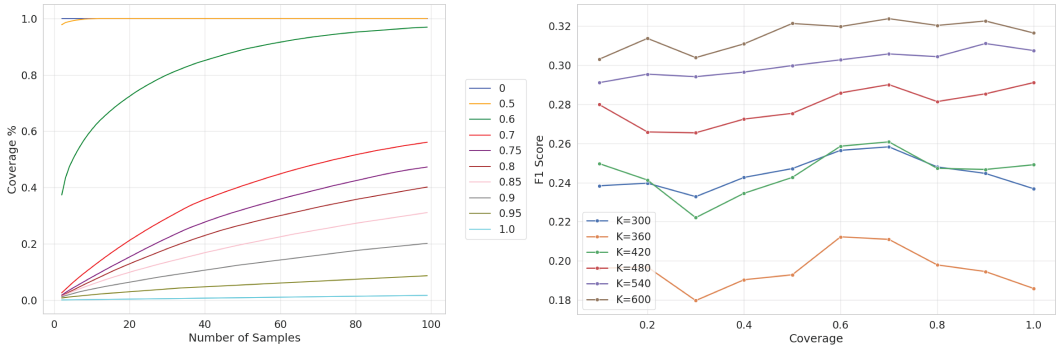

Figure 4: Empirical Results for the CrossNER dataset. (L) Coverage of data increases with k or when decreasing the similarity threshold. Colors correspond to the fixed similarity thresholds depicted in the legend. (R) Model accuracy as a function of coverage level for the sentiment analysis tasks. Performance peaks at a coverage level below 1.0.

and run the ACS procedure on the reduced instance. Once the optimal edge similarity threshold τ^* is identified on this subset, it is reused to construct the similarity graph and perform ACS on the *full* dataset. This approach significantly reduces computational cost while maintaining effective coverage.

D.1 Theoretical Verification

Formally, let $G = (V, E)$ be the similarity graph constructed on the full dataset, where edges are defined between points with similarity exceeding a threshold τ. Let $V' \subset V$ denote a uniformly random subsample of size N', and let $G' = (V', E')$ be the induced subgraph. For any subset $S \subset V$, we define the normalized coverage as the fraction of nodes in V that are neighbors of some node in S under threshold τ. We proceed to show that threshold tuning on the subsample generalizes well to the full dataset, in the following theorem.

Theorem D.1. *Let V be a finite set with $|V| = N$, fix a similarity threshold $\tau \in [0, 1]$, and fix an integer $k \geq 1$. For any $S \subseteq V$, define the (normalized) τ-coverage on the full dataset as*

$$Cov_\tau(S; V) := \frac{1}{|V|} \left| \{u \in V : \exists v \in S \text{ with } sim(u, v) \geq \tau\} \right|.$$

Sample $V' \subseteq V$ uniformly at random among all subsets of size $|V'| = N'$ (without replacement) and define the empirical coverage on the subsample as

$$Cov_\tau(S; V') := \frac{1}{|V|} \left| \{u \in V' : \exists v \in S \text{ with } sim(u, v) \geq \tau\} \right|.$$

Then, for any $\delta \in (0,1)$, with probability at least $1-\delta$ over V',

$$\sup_{S\subseteq V:|S|=k} |\mathit{Cov}_\tau(S;V') - \mathit{Cov}_\tau(S;V)| \le \sqrt{\frac{\log\left(2\binom{n}{2}/\delta\right)}{2N'}}.$$

This result further gives the following corollary.

Corollary D.2. *Let*

$$\mathit{OPT}_\tau(V) := \max_{S\subseteq V:|S|=k} \mathit{Cov}_\tau(S;V), \quad \mathit{OPT}_\tau(V') := \max_{S\subseteq V:|S|=k} \mathit{Cov}_\tau(S;V')$$

Then, on the same probability event as Theorem D.1, we have

$$|\mathit{OPT}_\tau(V') - \mathit{OPT}_\tau(V)| \le \sqrt{\frac{\log\left(2\binom{n}{2}/\delta\right)}{2N'}}.$$

We proceed to prove the results. Throughout this secton, we fix $\tau \in [0,1]$ and $k \ge 1$. For any $S \subseteq V$, define the indicator function

$$f_S(u) := \mathbf{1}\left[\exists v \in S : \mathrm{sim}(u,v) \ge \tau\right]$$

This allows us to rewrite

$$\mathrm{Cov}_\tau(S;V) = \frac{1}{N}\sum_{u\in V} f_S(u), \quad \mathrm{Cov}_\tau(S;V') = \frac{1}{N}\sum_{u\in V'} f_S(u)$$

We first prove the following lemma.

Lemma D.3. *For any fixed $S \subseteq V$,*

$$\mathbb{E}\left[\mathit{Cov}_\tau(S;V')\right] = \mathit{Cov}_\tau(S;V)$$

Proof. Observe that we can write

$$\mathrm{Cov}_\tau(S;V') = \frac{1}{N'}\sum_{u\in V} f_S(u)\mathbf{1}[u\in V']$$

Taking the expectations and using linearity, we directly obtain:

$$\mathbb{E}[\mathrm{Cov}_\tau(S;V')] = \frac{1}{N'}\sum_{u\in V} f_S(u)\mathbb{P}(u\in V')$$

Since V' is a uniform subset of V, $\mathbb{P}[u\in V'] = {}^{N'}\!/_{N}$ for each u. Hence

$$\begin{aligned}\mathbb{E}[\mathrm{Cov}_\tau(S;V')] &= \frac{1}{N'}\sum_{u\in V} f_S(u)\frac{N'}{N} \\ &= \frac{1}{N}\sum_{u\in V} f_S(u) \\ &= \mathrm{Cov}_\tau(S;V)\end{aligned}$$

□

We next invoke a Hoeffding-style concentration bound.

Lemma D.4. *For any fixed $S \subseteq V$ and any $t > 0$,*

$$\mathbb{P}\left[|\mathit{Cov}_\tau(S;V') - \mathit{Cov}_\tau(S;V) \ge t\right] \le 2\exp(-2N't^2).$$

Proof. We create V' by choosing a uniformly random permutation π of V and selecting $V' = \{\pi(1), ..., \pi(N')\}$. Define $Y := \sum_{i=1}^{N'} f_S(\pi(i))$ such that $\text{Cov}_\tau(S; V') = Y/N'$. Let $\mathcal{F}_j := \sigma(\pi(1), ..., \pi(j))$ and define the Doob martingale

$$M_j := \mathbb{E}[Y|\mathcal{F}_j],\ j = 0, 1, ..., N'.$$

Thus, $M_0 = \mathbb{E}[Y]$ and $M_{N'} = Y$.

We first show that $|M_j - M_{j-1}| \leq 1$ for all j. Fix j and condition on $\mathcal{F}_{j-1}$. Under this conditioning, $\pi(j)$ is uniform over the remaining $N - (j-1)$ elements. Revealing $\pi(j)$ reveals the value of $f_S(\pi(j))$ and further changing this value can alter Y by at most 1 (since the variables in this summation are bounded in $\{0, 1\}$). Hence, after the conditional expectation we have a change of at most 1.

Azuma-Hoeffding now gives, for any $a > 0$,

$$\begin{aligned}\mathbb{P}[|Y - \mathbb{E}[Y]| \geq a] &= \mathbb{P}[|M_{N'} - M_0| \geq a] \\ &\leq 2\exp\left(-\frac{2a^2}{\sum_{j=1}^{N'} 1^2}\right) \\ &= 2\exp\left(-\frac{2a^2}{N'}\right).\end{aligned}$$

Setting $a = N't$ and dividing by N' we obtain

$$\mathbb{P}\left[|Y/N' - \mathbb{E}[Y]/N'| \geq t\right] \leq 2\exp(-2N't^2).$$

Lastly, we have thta $Y/N' = \text{Cov}_\tau(S; V')$ and by Lemma D.3 we have $\mathbb{E}[Y]/N' = \text{Cov}_\tau(S; V)$ which completes the lemma. □

We can now prove the main theorem.

Proof of Theorem D.1. Set $\varepsilon := \sqrt{\frac{\log\left(2\binom{n}{2}/\delta\right)}{2N'}}$. Then $2\binom{N}{k}e^{-2N'\varepsilon^2} = 2\binom{N}{k}e^{-\log(2\binom{N}{k}/\delta)} = \delta$. Applying Lemma D.4 with this ε gives the desired bound. □

Proof of Corollary D.2. On the event of Theorem D.1, for every S with $|S| = k$,

$$\text{Cov}_\tau(S; V) - \varepsilon \leq \text{Cov}_\tau(S; V') \leq \text{Cov}_\tau(S; V) + \varepsilon.$$

Taking maxima over S on both sides yields the desired bound in terms of OPT_τ. □

D.2 Empirical Verification

To further validate this claim empirically, we conducted a series of experiments across the datasets used in the main text (sentiment analysis, relation extraction, and named entity recognition). In each setting, we selected a random subset of the data at varying proportions, ranging from very small to nearly the full dataset. For each subset, we used binary search to identify the threshold τ^* such that the greedy ACS procedure on the subset achieved a fixed target of 90% coverage with k examples. We then applied this same threshold τ^* to construct the similarity graph for the full dataset and ran the greedy max coverage to select a size-K subset, measuring the resulting coverage over all data points.

Figure 10 summarizes the results. Each plot corresponds to a different dataset (SST2, FewRel, or CrossNER). The x-axis represents the fraction of the dataset used to compute the optimial threshold, and the y-axis shows the actual coverage obtained on the full dataset using that threshold. A shaded band indicates an ε-envelope centered at the target coverage of 90% where ε is set to be 5×10^{-3}. Across all settings, we observe that even small subsamples, often less than 20% of the full dataset, yield thresholds that generalize well. As the sample size increases, the coverage rapidly converges to the target, and variance remains low throughout.

These results provide strong empirical support for the scalable ACS approach. By selecting a threshold on a small, randomly drawn subset, we can achieve nearly identical coverage behavior on the

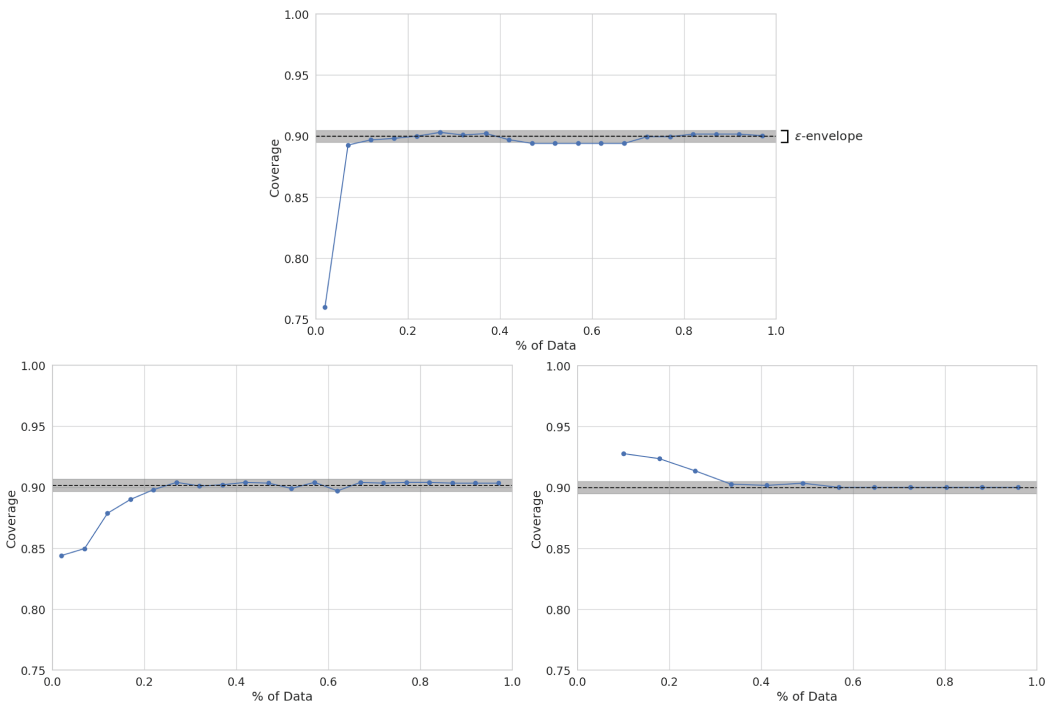

Figure 5: Coverage transfer from subsample to full dataset. Each point corresponds to a threshold τ^* optimized on a random subset of a given size and evaluated for coverage on the full dataset. The gray band denotes a small tolerance range around the 90% target. Results show threshold transfer achieves accurate and stable coverage across various dataset sizes. (Top Left) SST2, (Bottom Left), CrossNER, (Bottom Right) FewRel.

full dataset, enabling efficient and accurate training data selection in large-scale scenarios without repeated expensive graph construction or threshold tuning.

We note that the above experiments, in line with Theorem D.1 do not impose any max degree constraints on the similarity graph. We demonstrate that even when such constraints are imposed, the scalability of optimal threshold remains. In Figure 7, we again impose the max degree constraint of $2 \cdot c \cdot N/k$ and set a target coverage of 0.5.

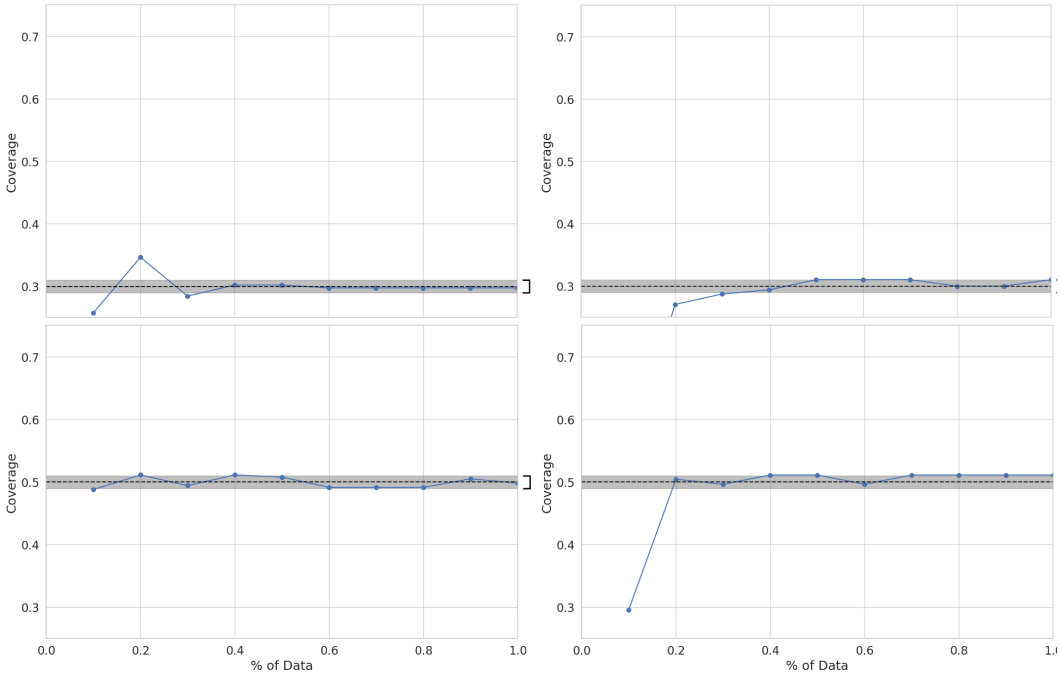

Figure 6: Coverage transfer from subsample to full dataset. Each point corresponds to a threshold τ^* optimized on a random subset of a given size and evaluated for coverage on the full dataset. The gray band denotes a small tolerance range around the 30% and 50% targets. Results show threshold transfer achieves accurate and stable coverage across various dataset sizes. (Left) SST2, and (Right) CrossNER.

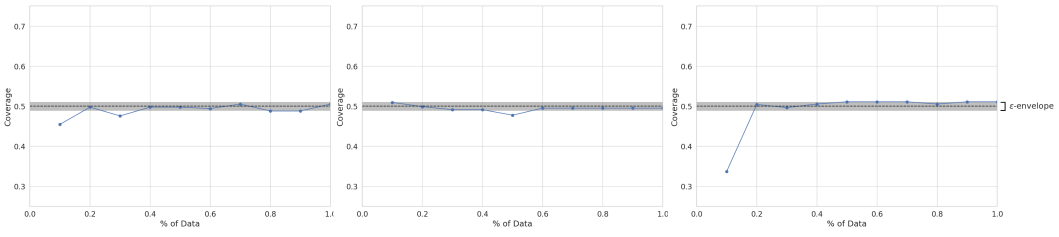

Figure 7: Coverage transfer from subsample to full dataset. Each point corresponds to a threshold τ^* optimized on a random subset with max degree constraint of a given size and evaluated for coverage on the full dataset. The gray band denotes a small tolerance range around the 50% target. Results show threshold transfer achieves accurate and stable coverage across various dataset sizes. (Left) SST2, (Middle) FewRel and (Right) CrossNER.