Loss-Based Active Learning for Neural Abstractive Summarization

Anonymous ACL submission

Abstract

Fine-tuning abstractive summarization models requires high-quality annotated data. However, obtaining such corpora is expensive and time-consuming, as it requires human annotators to read and comprehend long documents to create accurate summaries. Active learning mitigates this issue by selecting only the most informative instances for annotation, allowing models to achieve competitive results with significantly fewer labels. However, the application of active learning to summarization remains under-explored, and existing studies often suffer from instability and significant computational bottlenecks. To overcome these challenges, we propose LOBSTER (LOss-BaSed acTivE leaRning), a novel active learning framework designed specifically for abstractive summarization. LOBSTER improves performance by prioritizing unlabeled instances semantically similar to the model's current high-loss training examples, enabling the model to explicitly correct its specific weaknesses. Our empirical evaluation across three benchmark datasets and two summarization backbone models demonstrates that LOBSTER consistently matches or outperforms current state-of-the-art approaches while achieving a query selection speedup of up to $665\times$.

1 Introduction

Abstractive summarization aims at generating concise and fluent summaries that capture the main ideas of a piece of text by producing novel words and phrases beyond direct copying from the source (See et al., 2017; Zhang et al., 2020). State-of-the-art performance has been achieved by pre-trained large language models (LLMs) (Lewis et al., 2020; Zhang et al., 2020; Zaheer et al., 2020), based on the Transformers architecture (Vaswani et al., 2017). Nevertheless, fine-tuning LLMs for abstractive summarization is costly, since producing high-quality summaries requires human annotators to carefully read the full document and identify important information (Tsvigun et al., 2022; Perlitz et al., 2023).

Active learning addresses this challenge by iteratively selecting the most informative examples for annotation. This enables models to achieve strong performance with fewer training instances, thereby reducing the overall cost of obtaining labeled data (Settles, 2012). While recent efforts to apply active learning to abstractive summarization have shown promise, existing methods suffer from critical limitations. Uncertainty-based strategies are computationally demanding and tend to select outliers, as these instances often exhibit high uncertainty (Gidiotis and Tsoumakas, 2024). Conversely, diversity-based approaches can be unstable since they may overemphasize dense regions of the data space (Tsvigun et al., 2022), and more recent methods rely on external LLM systems, which can introduce additional cost and dependency (Li et al., 2024).

This work introduces a novel technique that strikes a balance between summarization performance and computational efficiency. We demonstrate that the cross-entropy loss of current labeled examples provides a useful signal for guiding the model toward the specific data it requires to improve. Building on this insight, we present LOBSTER (LOss-BaSed acTivE leaRning), a method that integrates density-based diversity with a mechanism that projects the hardness of labeled examples to unlabeled ones based on their semantic similarity. Our experiments demonstrate that LOBSTER matches or surpasses state-of-the-art performance while being significantly faster than uncertainty-based strategies and more stable than methods relying solely on diversity. To the best of our knowledge, we are the first to use sequence-to-sequence cross-entropy loss of labeled data to guide the selection of unlabeled data in abstractive summarization.

The main contributions of this work are as follows:

- We propose a hybrid selection strategy that explicitly targets the weaknesses of the current model while maintaining data diversity.
- We conduct extensive experiments across three benchmark datasets and two model backbones, demonstrating the effectiveness of LOBSTER while highlighting the inherent instability of diversity-based methods and the high latency of uncertainty-based strategies.
- We present a detailed data efficiency analysis revealing the surprising competitiveness of random sampling, under large annotation budgets.

2 Related Work

2.1 Abstractive Summarization

The introduction of neural sequence-to-sequence models (Sutskever et al., 2014) and the attention mechanism (Bahdanau et al., 2015) established the foundations for modern abstractive summarization. Building on these advances, the Transformer architecture (Vaswani et al., 2017) enabled large-scale pre-trained language models such as BART (Lewis et al., 2020), T5 (Raffel et al., 2020) and PEGASUS (Zhang et al., 2020), which set the state of the art. Recent work introduced model adaptations for long or multi-document inputs (Khanna et al., 2022; Xiao et al., 2022), while instruction-tuned LLMs demonstrated strong few-shot and zero-shot capabilities across various tasks, including abstractive summarization, without task-specific fine-tuning (Achiam et al., 2023; Chung et al., 2024).

2.2 Active Natural Language Generation

Within the field of NLP, active learning has been extensively studied and applied to discriminative tasks such as text classification (Ein-Dor et al., 2020; Margatina et al., 2021; Yu et al., 2022) and sequence tagging (Shen et al., 2017; Radmard et al., 2021). However, its application to generative tasks has received comparatively little attention (Perlitz et al., 2023). Existing active learning approaches for natural language generation are mainly categorized into uncertainty-based, diversity-based, and hybrid methods (Zhang et al., 2022).

Uncertainty-based methods. These methods select examples for annotation where the model is least confident in its predictions (Settles, 2012). Gidiotis and Tsoumakas (2022) leverage Bayesian approximation to model uncertainty, by adapting the Monte Carlo BLEU variance metric (Xiao et al., 2020) to summarization. They subsequently apply this uncertainty estimation framework to active learning via their Bayesian Active Summarization (BAS) approach (Gidiotis and Tsoumakas, 2024). Documents with the highest variance are selected for annotation, and the model is iteratively retrained on the labeled set. Empirical results indicate that this approach achieves performance comparable to full-data training with only a few hundred examples under specific experimental settings. Despite these gains, however, purely uncertainty-based strategies in abstractive summarization can be sensitive to data quality, often remaining susceptible to selecting noisy or uninformative examples (Tsvigun et al., 2022; Li et al., 2024).

Diversity-based methods. The goal of these methods is to select examples that are both diverse with respect to the labeled set while being representative of the data distribution (Kim et al., 2006). Tsvigun et al. (2022) propose In-Domain Diversity Sampling (IDDS), which selects instances that differ from the labeled set while still remaining close to the unlabeled pool, balancing diversity with domain relevance. Building on this idea, Li et al. (2024) introduce LLM-Determined Curriculum Active Learning (LDCAL), which leverages LLMs to rate instance difficulty and guide training. They also propose *certainty gain maximization*, a complementary strategy that selects instances whose distribution aligns closely with the overall data.

Hybrid methods. Recent work has shown that combining uncertainty and diversity methods, can outperform uncertainty-only and diversity-only methods in multiple settings (Azeemi et al., 2025; Giouroukis et al., 2025). Azeemi et al. (2025) introduce Hybrid Uncertainty and Diversity Sampling (HUDS), a hybrid strategy for neural machine translation that combines normalized negative log-likelihood for uncertainty with cosine distance from cluster centroids for diversity. For abstractive summarization, Diversity and Uncertainty Active Learning (DUAL) (Giouroukis et al., 2025) first selects a compact, diverse yet representative subset via IDDS, and then measures uncertainty within that subset using BAS.

3 Methodology

Under cross-entropy training, high-loss examples tend to generate larger gradients and thus dominate parameter updates, a property reflected in hard example mining (Shrivastava et al., 2016) and boosting methods (Freund and Schapire, 1997). We exploit this property in the active learning setting by directing supervision toward unlabeled instances that are semantically aligned with high-loss training samples, concentrating annotation budget in regions where parameter updates have greater impact. Our core hypothesis is that an active learning strategy can improve performance by prioritizing unlabeled instances that are semantically similar to the model's current high-loss training samples. By explicitly targeting the unlabeled *semantic counterparts* of these hard examples, the model is driven to correct its specific weaknesses.

To achieve this, we introduce LOBSTER (LOss-BaSed acTivE leaRning), a 3-stage sample acquisition strategy that first identifies high-loss examples in the labeled set, then applies density-based filtering to construct a representative candidate pool, and finally selects samples that are semantically similar to the difficult instances without the need for the expensive autoregressive decoding used by uncertainty methods. The complete workflow of the LOBSTER strategy is illustrated in Figure 1.

3.1 Problem Formulation

We define the active learning task for abstractive text summarization as follows. Let $\mathcal{U}$ represent the large unlabeled pool of documents, and $\mathcal{L}$ represent the initial small labeled set. At each active learning cycle t, an acquisition function utilizes the current model θ_t to select a batch of B instances from $\mathcal{U}$ to be annotated. The goal is to maximize summarization performance on a held-out test set while minimizing the total number of annotated samples.

3.2 Loss-Based Active Learning

LOBSTER is a hybrid active learning method that combines loss-guided sampling with IDDS in order to maximize performance. It consists of three stages which we introduce below.

Stage 1: Loss-Based Hard Example Selection. Unlike classification tasks, where errors correspond to strict binary failures, difficulty in abstractive summarization is better reflected by generation loss. High-loss examples are therefore more likely to correspond to error-prone regions of the data distribution. The first stage of LOBSTER aims to diagnose these regions. Unlike uncertainty sampling, which estimates uncertainty on unlabeled data, we explicitly measure generation difficulty on the *labeled* data. We first compute the token-level cross-entropy loss for all samples in the current labeled set $\mathcal{L}$. For a given source document x and target summary y of length T, the loss is defined as the average negative log-likelihood over the sequence:

$$L(x, y) = -\frac{1}{T}\sum_{t=1}^{T} \log P(y_t \mid y_{<t}, x; \theta) \quad (1)$$

We then identify the set of *hard examples*, $\mathcal{H}$, defined as the top-k examples with the highest loss:

$$\mathcal{H} = \{x_i \in \mathcal{L} \mid L(x_i, y_i) \geq \tau_k\} \quad (2)$$

where τ_k corresponds to the loss of the k-th example in $\mathcal{L}$ when sorted by loss in descending order. This acts as a dynamic threshold that adapts to the model's current performance, rather than a fixed predefined threshold. These high-loss instances serve as *semantic anchors*, representing the specific linguistic patterns or domains where the model is currently underperforming.

Stage 2: Representative Candidate Filtering. We observed that retrieving candidates purely via embedding similarity to the hard example set frequently produced clusters of highly correlated, near-duplicate examples. Therefore, we employ IDDS (Tsvigun et al., 2022) as a diversity-aware pre-filter to ensure the candidate pool covers broad, representative regions of the data distribution. Crucially, this mitigates the risk of selecting noisy outliers (a common failure mode in uncertainty-based active learning (Tsvigun et al., 2022)). Furthermore, while density-based methods are known to oversample dense regions when used as a final selector (Li et al., 2024), we avoid this issue by using IDDS strictly to construct a large candidate pool ($U_{rep} \gg B$) rather than for final budget allocation. Filtering is separated from the final selection in Stage 3, ensuring the candidate pool is representative and distinct from already labeled data, effectively mitigating the risks of both outlier selection and dense-region oversampling.

Stage 3: Semantic Hardness Projection. The core contribution of LOBSTER is the explicit targeting of high-loss regions, i.e., difficult regions

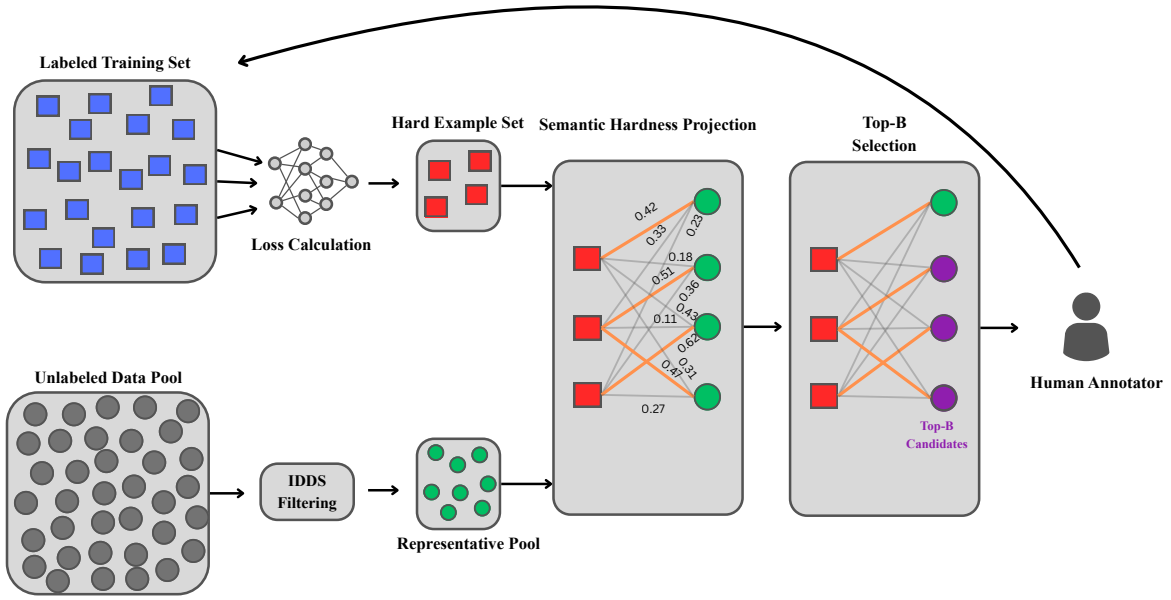


Figure 1: The three-stage pipeline of LOBSTER. (1) **Loss-based Hard Example Selection** identifies "Hard Example" set $\mathcal{H}$ (red squares) via high cross-entropy loss to serve as semantic anchors. (2) **Representative Candidate Filtering** uses IDDS to distill a pool of representative candidates (green circles) from the unlabeled set $\mathcal{U}$. (3) **Semantic Hardness Projection** computes the Max-Similarity Score $S(x_u)$ (Eq. 3) using Sentence-BERT embeddings. This helps to identify a "semantic twin" for each hard example, forming the final Top-B batch for annotation.

of the data. We assume that the optimal sample to annotate is the *semantic twin* of a known hard example.

For every document x_u in the representative candidate pool $\mathcal{U}_{rep}$, we compute a maximum similarity score $S(x_u)$ against the set of hard examples $\mathcal{H}$ using Sentence-BERT (Reimers and Gurevych, 2019) embeddings $\phi(\cdot)$:

$$S(x_u) = \max_{x_h \in \mathcal{H}} \left(\frac{\phi(x_u) \cdot \phi(x_h)}{\|\phi(x_u)\| \|\phi(x_h)\|} \right) \quad (3)$$

Crucially, we use the $\max$ operator rather than the average. We do not want documents that are broadly related to the hard example set, rather we are searching for instances that are similar to specific hard examples. The final batch is constructed by selecting the top B candidates with the highest $S(x_u)$.

4 Experimental Setup

4.1 Datasets

Following standard benchmarks in the field, we evaluated our active summarization models using three widely used datasets representing diverse summarization styles. AESLC contains short emails with their subject lines as summaries (Zhang and Tetreault, 2019). XSum contains BBC news articles with their single-sentence summaries, requiring high-level abstraction (Narayan et al., 2018). Finally, the CNN/DailyMail dataset consists of news articles accompanied by multi-sentence, bullet-point summaries, providing a benchmark for longer-form abstractive summarization (Hermann et al., 2015; Nallapati et al., 2016; See et al., 2017). Table 2 in Appendix A reports the size of the training and test sets, alongside the average token lengths for both source documents and reference summaries. All datasets consist entirely of English text. We cleaned all datasets by removing noisy instances, such as empty strings and duplicates. Additionally, we filtered the XSum and CNN/DailyMail datasets by excluding documents with fewer than 10 tokens and summaries with fewer than 3 tokens. Finally, due to the high computational cost of the uncertainty-based baselines across multiple active learning cycles, we evaluated our models on a random subset of 1,000 instances from the test sets.

4.2 Active Learning Setup

We adopted the conventional framework for active learning annotation, consistent with prior studies (Siddhant and Lipton, 2018; Shelmanov et al., 2021). We initialized the cycle by randomly sampling a seed set of 10 annotated instances. This warm-up phase is crucial for model-dependent strategies: for LOBSTER, the trained model is required to identify the initial set of hard examples; for strategies relying on uncertainty estimates, it is necessary to infer the initial uncertainty scores. In contrast, model-agnostic strategies like IDDS do not require this warm-up step. In each iteration, we select the top 10 examples from the unlabeled pool $\mathcal{U}$ based on the acquisition strategy's ranking. These examples are moved to the labeled pool $\mathcal{L}$, where we retrieve the ground-truth summaries to mimic human annotation. After every step, we fine-tune a summarization model from scratch on the updated $\mathcal{L}$ and evaluate it on the held-out test set. We repeat this loop for 15 iterations with a total budget of 150 instances and report the average performance over 5 independent runs.

4.3 Baselines

We compare our approach against the following baselines and state-of-the-art active summarization methods: random sampling, IDDS (Tsvigun et al., 2022), BAS (Gidiotis and Tsoumakas, 2024), and DUAL (Giouroukis et al., 2025). As discussed in Section 2, these methods represent a diverse spectrum of strategies, including diversity-based, uncertainty-based, and hybrid approaches. To ensure a fair comparison with previous studies, we used the code provided by the respective authors.

4.4 Implementation Details

We employed two pre-trained language models as the backbones of our experiments: BART-base (Lewis et al., 2020) and PEGASUS-large (Zhang et al., 2020). We utilized standard implementations for BART-base ($\sim$140M parameters) and PEGASUS-large ($\sim$568M parameters). We tuned the hyperparameters based on performance on a small, randomly sampled subset of the Gigaword dataset (Graff et al., 2003). We conducted all experiments using the HuggingFace Transformers library. The specific hyperparameters used for each model are detailed in Table 3 in Appendix A. All experiments were conducted on a Google Cloud Platform (GCP) virtual machine equipped with an NVIDIA L4 GPU. Our code is publicly available at GitHub.

4.5 Evaluation Metrics

We evaluate summary quality using ROUGE scores (Lin, 2004). Specifically, we compute ROUGE-1, ROUGE-2, and ROUGE-L, which measure the overlap of unigrams, bigrams, and longest common subsequences, respectively. While we report results for all three metrics, our analysis primarily focuses on ROUGE-1, as it provides a robust overall indication of content coverage. We also observed that ROUGE-2 and ROUGE-L exhibit performance trends consistent with ROUGE-1 across all experiments.

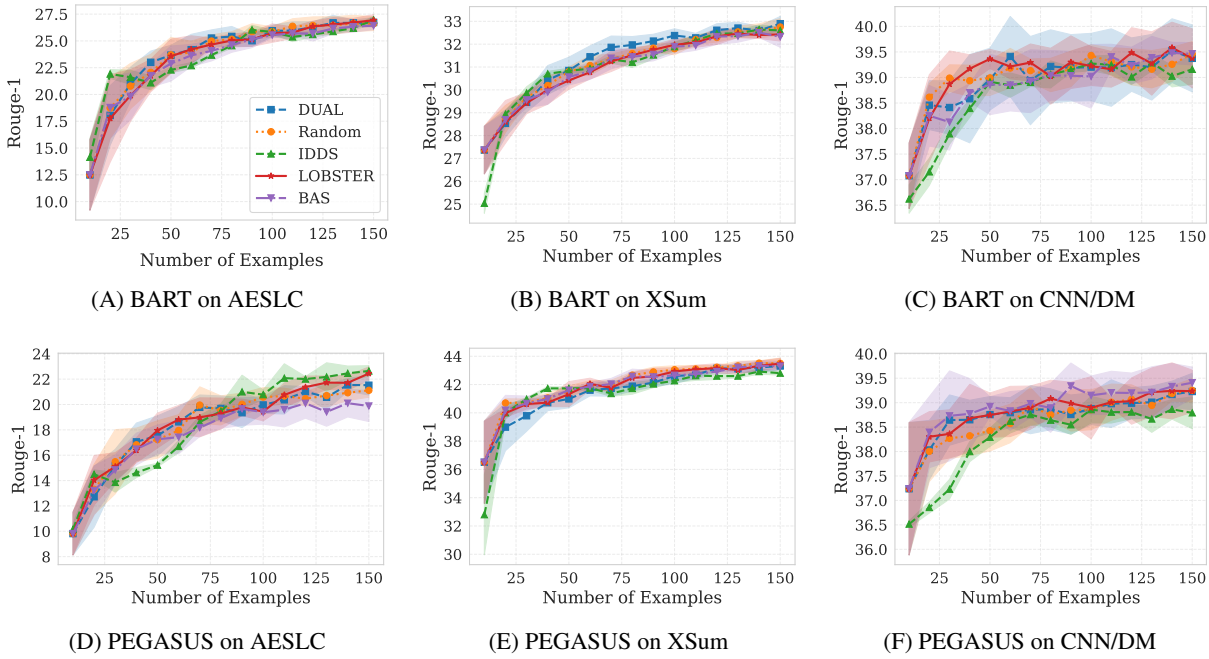
5 Results Analysis

5.1 Performance Comparison

We evaluate the summarization performance of LOBSTER against four strategies across three benchmark datasets and two backbone architectures. Figure 2 illustrates the ROUGE-1 learning curves on the test set across active learning iterations, where solid lines represent the mean score across 5 random seeds and shaded regions indicate the standard deviation. The full results at the final annotation budget are reported in Appendix B.

Competitive Effectiveness. Our analysis indicates that the efficacy of active learning strategies is often context-dependent, varying significantly across different datasets and backbone architectures. For instance, on the CNN/DM dataset, BAS achieves the top performance with the PEGASUS model. Conversely, on the XSum dataset using the BART backbone, DUAL emerges as the strongest performer. This variability suggests that standard active learning approaches are not universally optimal and are sensitive to the interaction between model architecture and summarization style. However, LOBSTER exhibits strong results, delivering performance that is consistently competitive with or superior to state-of-the-art baselines across these diverse experimental conditions. Specifically, on highly abstractive benchmarks such as XSum, LOBSTER achieves the highest ROUGE scores with the PEGASUS backbone, outperforming computationally expensive uncertainty-based approaches like BAS and DUAL.

Addressing the Cold Start Problem. A key finding is LOBSTER's ability to correct the inherent



(A) BART on AESLC (B) BART on XSum (C) BART on CNN/DM

(D) PEGASUS on AESLC (E) PEGASUS on XSum (F) PEGASUS on CNN/DM

Figure 2: Number of labeled instances vs. ROUGE-1. Comparison of active strategies using BART and PEGASUS backbones on AESLC, XSum, and CNN/DM datasets.

weaknesses of the model. Although LOBSTER builds upon the IDDS framework, it demonstrates superior performance. While IDDS is generally competitive, it exhibits notable instability in our experiments, particularly on the CNN/DM dataset and on AESLC with PEGASUS. As seen in Figures 2C, 2D, and 2F, IDDS suffers from a severe "cold start" problem, lagging significantly behind all other methods during the initial acquisition cycles. This initial instability is particularly critical because the core premise of active learning is to maximize performance under strict annotation budgets. In contrast, LOBSTER demonstrates more stable performance throughout the active learning process, as it avoids the initial cold start problem that IDDS faces.

5.2 Computational Efficiency

While LOBSTER demonstrates competitive results in all different scenarios, its primary contribution lies in its computational efficiency. Table 1 compares the average runtime (in seconds) for a single active learning selection iteration across different strategies, datasets and models.

Scalability and Runtime Analysis. Strategies relying on autoregressive decoding (e.g., BAS, DUAL) become prohibitively expensive as model and dataset sizes increase. These approaches estimate uncertainty by generating multiple distinct summaries for each candidate document via multiple stochastic forward passes (Monte Carlo dropout). Because these passes must be repeated at each active learning iteration to reassess the unlabeled pool, the computational overhead grows considerably with larger datasets and model scales. These strategies exhibit poor scalability, becoming increasingly impractical as model architectures scale up or as generation complexity increases.

Table 1: Average runtime (in seconds) for instance selection, for one active learning iteration, across different strategies. Comparison across backbones and datasets.

		Avg. Selection Time (s)			
Model	**Dataset**	BAS	DUAL	IDDS	LOBSTER
BART-base	AESLC	46.1	38.7	0.02	0.6
	XSum	66.6	63.7	0.4	0.8
	CNN/DM	70.2	69.2	0.3	0.8
PEGASUS-large	AESLC	221.1	113.6	0.01	1.0
	XSum	390.0	267.6	0.4	1.5
	CNN/DM	1064.2	699.4	0.4	1.6

In contrast, LOBSTER adopts a deterministic approach that shifts the computational burden from the expansive unlabeled pool to the much smaller labeled set. After the training phase, we compute the per-example cross-entropy loss using teacher forcing, which allows parallelized forward passes that are significantly faster than autoregressive token-by-token generation. Furthermore, LOBSTER replaces repeated model inference with ef-

ficient vector similarity computations in the embedding space, avoiding the expensive autoregressive bottleneck entirely and ensuring scalability for large-scale document pools and high-parameter models.

As demonstrated in Table 1, LOBSTER consistently and significantly outperforms both BAS and DUAL in terms of computational efficiency (we omit random sampling as its selection overhead is negligible). For example, with BART-Base on AESLC, LOBSTER reduces selection time to just 0.6 seconds, compared to 46.1 seconds for BAS and 38.7 seconds for DUAL. This efficiency gap widens as the model and document length increase; on CNN/DM with PEGASUS-Large, BAS and DUAL require 1064.2 and 699.4 seconds respectively, whereas LOBSTER completes the selection in only 1.6 seconds. This shows that LOBSTER achieves a speedup of 665× over BAS and 437× over DUAL in this configuration, effectively reducing the selection latency from nearly 18 minutes to near real-time (approaching the near-zero overhead of the model-agnostic IDDS) and enabling LOBSTER's viability for large-scale summarization applications. Crucially, LOBSTER achieves this near-zero latency without sacrificing the high summarization quality of uncertainty-based methods or suffering the cold-start instability of density-based baselines like IDDS.

5.3 Ablation Study: Impact of IDDS

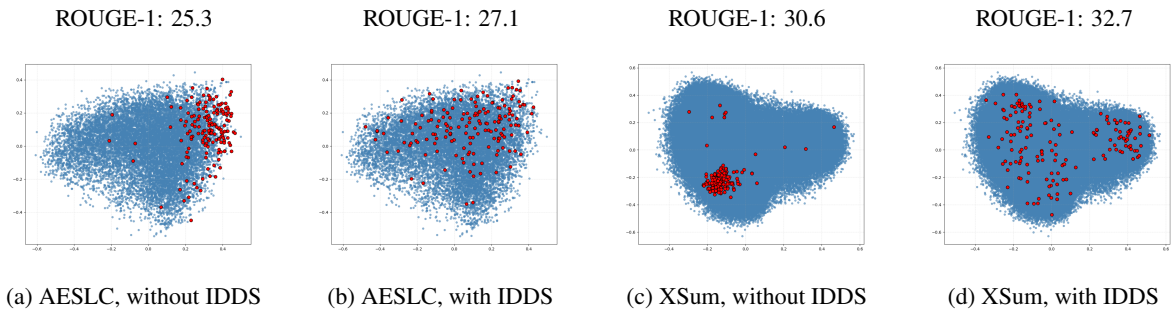
To isolate the contribution of the density-based pre-filtering (Stage 2), we removed the representative candidate pool ($\mathcal{U}_{rep}$) and applied the semantic hardness projection (Stage 3) directly to the entire unlabeled pool ($\mathcal{U}$). We evaluated this ablation on AESLC and XSum to assess robustness across different summarization styles.

Visualization of Semantic Collapse To visualize the behavior of the selection mechanism (Stage 3) without filtering, we applied PCA to Sentence-BERT embeddings. Figure 3 plots selected samples (red) against the unlabeled pool (blue), revealing a critical failure mode in the ablation baseline: LOBSTER without IDDS becomes trapped in specific embedding regions.This concentration reduces the model's ability to explore a broader range of linguistic patterns by repeatedly sampling from a narrow subset of the data. This effect is particularly evident in datasets like AESLC and XSum, which contain many near-duplicates examples. Relying solely on similarity to the "Hard Example" set retrieves nearly identical batches, wasting the annotation budget on redundant examples. In contrast, pre-filtering with IDDS mitigates this. As a representative pre-filter (Stage 2 in Section 3.2), IDDS ensures the candidate pool covers the broader underlying data distribution (U_{rep}) rather than a single cluster. This coverage constraint prevents the subsequent similarity-based selection from repeatedly picking local redundancies, ensuring each iteration selects diverse and informative examples.

5.4 Data Efficiency Setup

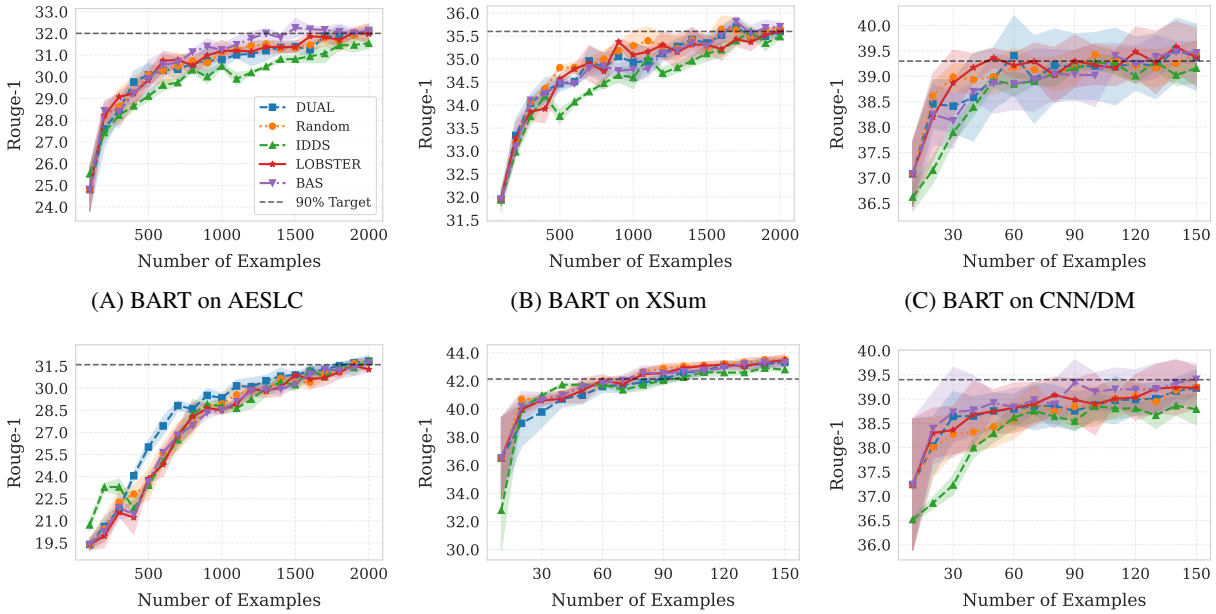
In preliminary experiments with a 150-sample budget, several configurations failed to reach 90% of full-dataset performance. Since such a small annotation budget is often unrealistic in practical settings, we expanded the budget for these underperforming setups (*BART/AESLC*, *PEGASUS/AESLC*, and *BART/XSum*) to 2,000 samples ($B = 100$ over 20 iterations). Configurations already achieving the 90% target (e.g., *CNN/DM*) required no expansion. Figure 4 and Appendix C detail the convergence rates and sample requirements needed to reach the 90% performance threshold, alongside a comprehensive runtime analysis of the selection process.

The Competitiveness of Random Sampling Table 5 in Appendix C reveals the surprisingly strong performance of the random sampling baseline.In several configurations, notably within the AESLC and XSum datasets, random sampling matches or even surpasses the sample efficiency of sophisticated active learning strategies. For instance, in the *BART-base/XSum* setup, random sampling reaches the 90% threshold using only 1,600 samples, outperforming all other approaches. Also, IDDS fails to reach the 90% performance threshold in half of the evaluated scenarios. While the uncertainty-based BAS strategy is the sole method to succeed on the *PEGASUS-large/CNN/DM* setup, its prohibitive computational latency makes it impractical for scalable deployment. LOBSTER successfully hits the target in 5 out of 6 configurations, providing a strong balance: it performs more reliably than IDDS while remaining faster than uncertainty methods. This indicates that within these specific benchmarks, when the annotation budget is sufficiently large, stochastic selection captures enough of the underlying data distribution to reach satisfactory performance. As a result, under these specific experimental conditions, complex active learning



(a) AESLC, without IDDS (b) AESLC, with IDDS (c) XSum, without IDDS (d) XSum, with IDDS

Figure 3: Impact of IDDS on Selection Diversity (PCA visualization of selected instances). Comparison of the proposed LOBSTER method with IDDS versus the ablation without IDDS. The IDDS module prevents semantic collapse by filtering redundant candidates, leading to higher ROUGE-1 scores.



(A) BART on AESLC (B) BART on XSum (C) BART on CNN/DM

(D) PEGASUS on AESLC (E) PEGASUS on XSum (F) PEGASUS on CNN/DM

Figure 4: Number of labeled instances vs. ROUGE-1. Comparison of active learning strategies using BART and PEGASUS backbones on AESLC, XSum, and CNN/DM datasets. The dashed horizontal line in each subplot denotes the 90% performance threshold of the full-dataset baseline.

methods do not provide a significant advantage over simply selecting random documents for abstractive text summarization.

6 Conclusions and Future Work

This work demonstrated that the cross-entropy loss of labeled examples can serve as an effective acquisition signal for active learning in abstractive summarization. By identifying the model's high-loss examples from the labeled set, our approach targets the specific areas where the model struggles. Building on this, we select the unlabeled "semantic twins" of those high-loss cases so the model is explicitly guided to correct its specific weaknesses. Furthermore, we showed that using a density-based pre-filter (IDDS) to find a representative candidate pool before final selection prevents semantic collapse. Extensive evaluation across three benchmark datasets and two backbone architectures confirms that LOBSTER matches or outperforms state-of-the-art baselines while remaining significantly faster. Finally, our data efficiency analysis indicates that random sampling remains surprisingly competitive under large annotation budgets. As the budget increases, random selection captures enough of the underlying data distribution to yield strong summarization performance. As future work, we plan to explore the applicability of LOBSTER to other sequence-to-sequence generation tasks beyond abstractive summarization, as well as its adaptation to modern LLMs.

Limitations

While our study demonstrates promising results, a few limitations remain. Our experiments are limited to English-language datasets (AESLC, XSum, and CNN/DailyMail), and therefore the effectiveness of LOBSTER for other languages remains unexplored. Languages with different morphology, syntax, or summarization styles may influence the behavior of embedding similarity and active learning selection strategies. Second, our evaluation relies primarily on ROUGE scores which, although widely used, primarily measure lexical overlap and may not fully capture the factual consistency or semantic quality of the generated summaries. Finally, our experiments simulate human annotation by retrieving ground-truth summaries from existing datasets rather than collecting new human annotations, which may not fully reflect real-world annotation scenarios.

Ethical Considerations

This work focuses on improving the efficiency of active learning for abstractive text summarization. The proposed method does not introduce new datasets, but it relies on pre-trained language models and existing benchmark corpora, which may contain biases present in their original training data. As a result, biases in the underlying models or datasets could influence the selection of instances during the active learning process and may propagate to downstream summarization systems. Future work could explore bias-aware acquisition strategies or incorporate fairness constraints when selecting samples for annotation.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Abdul Hameed Azeemi, Ihsan Ayyub Qazi, and Agha Ali Raza. 2025. To label or not to label: Hybrid active learning for neural machine translation. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3071–3082, Abu Dhabi, UAE. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, and 16 others. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Liat Ein-Dor, Alon Halfon, Ariel Gera, Eyal Shnarch, Lena Dankin, Leshem Choshen, Marina Danilevsky, Ranit Aharonov, Yoav Katz, and Noam Slonim. 2020. Active Learning for BERT: An Empirical Study. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7949–7962, Online. Association for Computational Linguistics.

Yoav Freund and Robert E Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139.

Alexios Gidiotis and Grigorios Tsoumakas. 2022. Should we trust this summary? Bayesian abstractive summarization to the rescue. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 4119–4131, Dublin, Ireland. Association for Computational Linguistics.

Alexios Gidiotis and Grigorios Tsoumakas. 2024. Bayesian active summarization. *Computer Speech & Language*, 83:101553.

Petros Stylianos Giouroukis, Alexios Gidiotis, and Grigorios Tsoumakas. 2025. Dual: Diversity and uncertainty active learning for text summarization. *Preprint*, arXiv:2503.00867.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English gigaword. *Linguistic Data Consortium, Philadelphia*, 4(1):34.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1693–1701, Cambridge, MA, USA. MIT Press.

Urvashi Khanna, Samira Ghodratnama, Diego Mollá, and Amin Beheshti. 2022. Transformer-based models for long document summarisation in financial domain. In *Proceedings of the 4th Financial Narrative Processing Workshop @LREC2022*, pages 73–78, Marseille, France. European Language Resources Association.

Seokhwan Kim, Yu Song, Kyungduk Kim, Jeong-Won Cha, and Gary Geunbae Lee. 2006. MMR-based

active machine learning for bio named entity recognition. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 69–72, New York City, USA. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Dongyuan Li, Ying Zhang, Zhen Wang, Shiyin Tan, Satoshi Kosugi, and Manabu Okumura. 2024. Active learning for abstractive text summarization via LLM-determined curriculum and certainty gain maximization. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8959–8971, Miami, Florida, USA. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. 2021. Active learning by acquiring contrastive examples. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 650–663, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Yotam Perlitz, Ariel Gera, Michal Shmueli-Scheuer, Dafna Sheinwald, Noam Slonim, and Liat Ein-Dor. 2023. Active learning for natural language generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9862–9877, Singapore. Association for Computational Linguistics.

Puria Radmard, Yassir Fathullah, and Aldo Lipani. 2021. Subsequence based deep active learning for named entity recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4310–4321, Online. Association for Computational Linguistics.

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

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Burr Settles. 2012. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Springer Cham, Cham. EBook published 31 May 2022.

Artem Shelmanov, Dmitri Puzyrev, Lyubov Kupriyanova, Denis Belyakov, Daniil Larionov, Nikita Khromov, Olga Kozlova, Ekaterina Artemova, Dmitry V. Dylov, and Alexander Panchenko. 2021. Active learning for sequence tagging with deep pre-trained models and Bayesian uncertainty estimates. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1698–1712, Online. Association for Computational Linguistics.

Yanyao Shen, Hyokun Yun, Zachary Lipton, Yakov Kronrod, and Animashree Anandkumar. 2017. Deep active learning for named entity recognition. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 252–256, Vancouver, Canada. Association for Computational Linguistics.

Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. 2016. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 761–769.

Aditya Siddhant and Zachary C. Lipton. 2018. Deep Bayesian active learning for natural language processing: Results of a large-scale empirical study. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages

2904–2909, Brussels, Belgium. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Akim Tsvigun, Ivan Lysenko, Danila Sedashov, Ivan Lazichny, Eldar Damirov, Vladimir Karlov, Artemy Belousov, Leonid Sanochkin, Maxim Panov, Alexander Panchenko, Mikhail Burtsev, and Artem Shelmanov. 2022. Active learning for abstractive text summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5128–5152, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Tim Z. Xiao, Aidan N. Gomez, and Yarin Gal. 2020. Wat zei je? detecting out-of-distribution translations with variational transformers. *Preprint*, arXiv:2006.08344.

Wen Xiao, Iz Beltagy, Giuseppe Carenini, and Arman Cohan. 2022. PRIMERA: Pyramid-based masked sentence pre-training for multi-document summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5245–5263, Dublin, Ireland. Association for Computational Linguistics.

Yue Yu, Lingkai Kong, Jieyu Zhang, Rongzhi Zhang, and Chao Zhang. 2022. AcTune: Uncertainty-based active self-training for active fine-tuning of pretrained language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1422–1436, Seattle, United States. Association for Computational Linguistics.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, volume 33, pages 17283–17297. Curran Associates, Inc.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

Rui Zhang and Joel Tetreault. 2019. This email could save your life: Introducing the task of email subject line generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 446–456, Florence, Italy. Association for Computational Linguistics.

Zhisong Zhang, Emma Strubell, and Eduard Hovy. 2022. A survey of active learning for natural language processing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6166–6190, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Appendix

A Dataset Statistics and Model Hyperparameters

In this section, we provide additional details regarding the datasets and models used in our experiments. Table 2 reports dataset statistics, including the number of instances and the average lenghts of source documents and summaries for both training and test splits. Table 3 provides the hyperparameters used for fine-tuning the backbone summarization models.

Table 2: Statistics for the datasets used in this work. We report the number of instances (# Ins.) and the average length of source documents (Doc. len.) and summaries (Sum. len.) for both training and test splits.

Dataset	Subset	# Ins.	Doc. len.	Sum. len.
Gigaword	Train	200	39.6	11.2
	Test	1K	38.3	10.4
AESLC	Train	14,436	115.3	4
	Test	1,906	114.6	4.1
XSum	Train	204,045	388.6	23.1
	Test	1K	384.3	23
CNN/DM	Train	287,113	692.4	51.6
	Test	1K	676.1	55.2

Table 3: Hyperparameters used for fine-tuning the backbone summarization models.

Hyperparameter	BART-base	PEGASUS-large
Learning Rate	$2e^{-5}$	$5e^{-5}$
Batch Size (Train)	16	8
Batch Size (Eval)	32	16
Num. Epochs	6	4
Optimizer	AdamW	AdamW
Weight Decay	0.028	0.03
Warmup Ratio	0.1	0.1
Beam Size	4	4
Min. Train Steps	350	200
Grad. Accumulation	1	1

In this work, we utilize standard, widely adopted public datasets (AESLC, XSum, CNN/DailyMail) and pre-trained models (e.g., BART, PEGASUS). These artifacts are well-established benchmarks within the NLP community and are used here strictly for their intended purpose of academic benchmarking and evaluating summarization performance. All datasets and models are used in accordance with their respective open-source licenses and research usage guidelines. Because our core contribution focuses on an algorithmic active learning strategy rather than novel dataset curation, we relied on the standard versions and original preprocessing of these datasets. Consequently, we did not perform additional data collection. We also did not apply further filtering for personally identifiable information or offensive content beyond the preprocesing steps already applied by the original dataset creators.

All experiments were conducted using publicly available implementations of the selected backbone models through the HuggingFace Transformers library. Training and evaluation procedures follow standard fine-tuning practices for abstractive summarization models. Hyperparameters were selected based on preliminary tuning experiments and remain consistent across all active learning strategies to ensure a fair comparison.

To promote reproducibility, all code created for this study, including the full experimental pipeline and the implementation of the proposed active learning method, is provided anonymously solely for the intended purpose of peer-review.

B Summarization Performance

Table 4 reports the summarization performance of all compared methods under a 150-instance annotation budget across the AESLC, XSum, and CNN/DailyMail datasets. Results are evaluated using ROUGE-1, ROUGE-2, and ROUGE-L.

C Detailed Data Efficiency Setup Results

In this section, we provide additional results for the data efficiency analysis discussed in Section 5.4. Table 5 reports the number of annotated samples required for each method to reach 90% of the full dataset performance across different datasets and backbone models. Additionally, Table 6 presents the average runtime required for instance selection during a single active learning iteration, highlighting the computational efficiency of the compared strategies.

These results are consistent with the trends discussed in the main text. In particular, Table 5 highlights the strong competitiveness of random sampling under larger annotation budgets, while LOBSTER also achieves strong and consistent performance across most setups. At the same time, LOBSTER provides a balance between performance reliability and computational efficiency compared to uncertainty-based approaches, as shown in Table 6.

Table 4: Summarization performance with a 150-instance annotation budget on AESLC, XSum, and CNN/DM datasets. The best performance in each column is highlighted in bold.

		AESLC			XSum			CNN/DM		
Model	**Method**	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
BART-base	Random	26.6	13.2	26.0	32.6	10.7	25.3	**39.5**	17.0	26.3
	BAS	26.5	13.1	25.8	32.3	10.6	25.1	**39.5**	17.0	**26.6**
	IDDS	27.0	12.5	25.4	32.6	10.6	25.0	39.2	16.5	26.2
	DUAL	26.9	13.2	26.0	**32.9**	**10.8**	**25.6**	39.4	**17.1**	**26.6**
	LOBSTER	**27.1**	**13.4**	**26.3**	32.5	10.6	25.3	39.4	17.0	26.4
PEGASUS-large	Random	21.1	9.9	20.0	43.4	20.1	34.9	39.2	17.0	26.3
	BAS	19.6	9.0	18.9	43.3	20.2	34.9	**39.4**	**17.1**	26.2
	IDDS	**22.6**	**11.0**	**21.5**	42.8	19.6	34.2	38.8	16.8	26.2
	DUAL	21.2	10.1	20.5	43.3	**20.3**	**35.0**	39.2	**17.1**	**26.4**
	LOBSTER	22.1	10.9	**21.5**	**43.5**	**20.3**	**35.0**	39.2	16.9	26.2

Table 5: Data Efficiency Analysis: Number of annotated samples required to reach 90% of the full-dataset performance. Bold values indicate the winning strategy for each configuration, and a dash (—) indicates that the configuration failed to reach the 90% target.

Model	**Dataset**	**Active Learning**				
		LOBSTER	**DUAL**	**BAS**	**IDDS**	**Rand.**
BART-base	AESLC	1900	1900	**1300**	—	2000
	XSum	2000	1700	1700	—	**1600**
	CNN/DM	**50**	60	110	100	100
PEGASUS-large	AESLC	**1800**	**1800**	2000	2000	**1800**
	XSum	**80**	90	**80**	100	**80**
	CNN/DM	—	—	**150**	—	—

Table 6: Data Efficiency Analysis: Average runtime (in seconds) for instance selection, for one active learning iteration, across different strategies. Comparison across backbones and datasets.

		Avg Selection time (s)			
Model	**Dataset**	BAS	DUAL	IDDS	LOBSTER
BART-base	AESLC	61.8	366.2	0.3	3.9
	XSum	96.1	560.5	2.8	12.1
	CNN/DM	70.2	69.2	0.3	0.8
PEGASUS-large	AESLC	164.6	618.1	0.3	12.8
	XSum	390.0	267.6	0.4	1.5
	CNN/DM	1064.2	699.4	0.4	1.6