

ATGen: A Framework for Active Text Generation

Akim Tsvigun¹ Daniil Vasilev² Ivan Tsvigun³ Ivan Lysenko² Talgat Bektleuov¹
Aleksandr Medvedev⁴ Uliana Vinogradova⁵ Nikita Severin³ Mikhail Mozikov⁶
Andrey Savchenko^{2,7} Rostislav Grigorev¹ Ramil Kuleev¹
Fedor Zhdanov⁸ Artem Shelmanov^{9*} Ilya Makarov^{1,6*}

¹Research Center of the Artificial Intelligence Institute, Innopolis University

²HSE University ³Independent Researcher ⁴T-Technologies ⁵Robotics Center

⁶AIRI ⁷SB-AI-Lab ⁸Royal Holloway University of London ⁹MBZUAI

a.tsvigun@innopolis.ru artem.shelmanov@mbzuai.ac.ae i.makarov@innopolis.ru

Abstract

Active learning (AL) has demonstrated remarkable potential in reducing the annotation effort required for training machine learning models. However, despite the surging popularity of natural language generation (NLG) tasks in recent years, the application of AL to NLG has been limited. In this paper, we introduce Active Text Generation (ATGen) – a comprehensive framework that bridges AL with text generation tasks, enabling the application of state-of-the-art AL strategies to NLG. Our framework simplifies AL-empowered annotation in NLG tasks using both human annotators and automatic annotation agents based on large language models (LLMs). The framework supports LLMs deployed as services, such as ChatGPT and Claude, or operated on-premises. Furthermore, ATGen provides a unified platform for smooth implementation and benchmarking of novel AL strategies tailored to NLG tasks. Finally, we present evaluation results for state-of-the-art AL strategies across diverse settings and multiple text generation tasks. We show that ATGen reduces both the effort of human annotators and costs associated with API calls to LLM-based annotation agents. The code of the framework is available on GitHub¹ under the MIT license. The video presentation is available at <http://atgen-video.nlpresearch.group>

1 Introduction

Natural language generation (NLG) has witnessed significant advancements in recent years, with the emergence of large language models (LLMs) such as o3 (OpenAI), Claude-4-Opus (Anthropic, 2025), DeepSeek-R1 (DeepSeek-AI et al., 2025), and others. These models have achieved remarkable performance across various NLG tasks, including reasoning, neural machine translation, and summarization. However, for tasks that require deep domain

knowledge, such as text generation in medical or law domains, even the most powerful LLMs are not capable of generating responses of adequate quality (Moëll, 2024). Hence, for such tasks, the availability of annotated datasets still remains a critical bottleneck. Moreover, due to latency constraints and memory limitations, real-world applications often require the deployment of low-resource models. Such models often exhibit low performance without task-specific fine-tuning, further emphasizing the need for annotated data.

Recently, automatic labeling methods have been introduced to alleviate the workload of human annotators by utilizing LLMs for labeling in instruct-mode (Honovich et al., 2023; Wang et al., 2023). Nonetheless, these techniques are not universally applicable, as current LLMs may struggle to generate high-quality annotations for domain-specific tasks and datasets. Querying the most powerful LLMs, such as o3 or Claude-4-Opus, incurs substantial costs, rendering large-scale data annotation prohibitively expensive.

Active learning (AL) is a promising approach to addressing the annotation bottleneck in machine learning. By strategically selecting for labeling the most informative instances, AL aims to maximize the model performance while minimizing the annotation effort (Settles, 2009). Instances are selected iteratively by batches, and after labeling each batch, the new instances are used to update an ML model, which in its turn is used to select another batch. AL in text and token classification tasks for Transformer-based models allows reducing the number of annotations by 3-5 times compared to random selection of instances while maintaining the same level of performance (Shelmanov et al., 2021; Margatina et al., 2021). In the era of LLM-powered annotation, AL emerges as a powerful tool – not only streamlining human effort but also reducing the total cost of LLM API calls for automatic data labeling.

¹<https://github.com/Aktsvigun/atgen>

* – equal contribution

Another promising direction in this area is experimental design (ED; [Bhatt et al. \(2024a\)](#)). In this approach, instances for annotation are selected once before training the model. This helps mitigate the significant overhead in AL ushered from training a model and querying samples to label on each iteration. ED also allows for parallelizing the labeling procedure. It is especially beneficial when humans serve as annotators because it eliminates the costs associated with the latency of training a model and performing an AL query on each iteration. However, to select new instances, ED utilizes neither labels obtained during the annotation process nor, consequently, the model knowledge after fine-tuning on the already annotated instances. This can potentially degrade its benefits compared to those of AL. For the sake of simplicity, for the remainder of the paper, we subsume ED approaches under the umbrella term AL since ED can be considered a particular case of AL.

Although there exist many NLP-oriented AL frameworks, they primarily focus on classification and sequence labeling tasks ([Lin et al., 2019](#); [Tsvigun et al., 2022b](#); [Schröder et al., 2023](#)). Furthermore, launching an AL cycle with modern LLMs requires parameter-efficient tools for fine-tuning (PEFT) and support for efficient LLM inference. Despite the recent progress in AL strategies for NLG tasks ([Tsvigun et al., 2022a](#); [Xia et al., 2024](#); [Azeemi et al., 2025](#)), there is currently no unified framework to evaluate these strategies in unified settings. Finally, given the remarkable performance of modern LLMs, powerful models can often effectively replace human annotators for labeling data on many relatively simple tasks ([Golde et al., 2023](#); [Peng et al., 2023](#)).

To fill the aforementioned gaps, we introduce Active Text Generation (ATGen) – a comprehensive framework that enables AL annotation in NLG tasks. ATGen aims to democratize active learning for text generation by making it accessible to users regardless of their expertise level in these topics. With just a few lines of code, users can initiate AL-empowered data annotation with human or LLM-based annotators. For researchers, the framework offers a unified platform for developing and benchmarking novel active learning strategies, thereby fostering further innovation in the field.

The main contributions of our framework can be summarized as follows:

- A collection of state-of-the-art AL and ED

strategies for text generation implemented with unified program interfaces.

- A demo web application that allows performing AL annotation for NLG tasks, supporting both manual labeling and automatic labeling via proprietary or open-source LLM-based annotation agents.
- A benchmarking platform for rigorous evaluation of AL strategies in NLG tasks.
- Experimental evaluation demonstrating that AL substantially reduces annotation time for manual labeling and total costs for LLM API calls in automatic annotation scenarios.

2 Related Work

2.1 AL Selection Strategies in NLP

Non-generative NLP tasks. AL has been widely explored for non-generative NLP tasks ([Yuan et al., 2020](#); [Margatina et al., 2021](#); [Shelmanov et al., 2021](#); [Liu et al., 2021](#); [Tsvigun et al., 2022c](#)). Most prominent solutions substantially outperform random sampling, emphasizing the benefits of AL, as shown by [Schröder et al. \(2022\)](#). Particularly, for text classification and token classification tasks, enabling AL allows reaching the same model quality with a 3-6 times reduced budget for annotation.

Technically, some of these strategies can be reused for generative tasks; however, their performance in these tasks is questionable. For example, least confidence ([Lewis and Gale, 1994](#)), prediction entropy ([Roy and McCallum, 2001](#)), and Coreset strategies ([Sener and Savarese, 2018](#)) were shown to substantially outperform random sampling in text classification ([Schröder et al., 2022](#); [Tsvigun et al., 2022b](#)). However, their extension in generative tasks does not improve the quality obtained through random sampling and can even lead to lower results ([Tsvigun et al., 2022a](#); [Perlitz et al., 2023](#)). Therefore, application of such strategies to NLG tasks requires careful evaluation in various settings before usage.

Text generation tasks. Recently, several AL strategies tailored to NLG tasks have emerged. **TE-delfy** ([Zhao et al., 2020](#)), introduced for the NMT task, combines uncertainty-based token entropy (TE) and model-agnostic decay logarithm frequency (delfy). **BLEUVar**, proposed by [Xiao et al. \(2020\)](#), considers documents as vectors and employs the BLEU score ([Papineni et al., 2002](#))

to compute the dissimilarity between them. It selects for annotation texts that exhibit the highest variability in BLEU scores across multiple stochastic generations. **NGF-SMP** (Hu and Neubig, 2021) selects frequent phrases that are under-represented in the labeled data. **HUDS** (Azeemi et al., 2025) combines uncertainty-based and metric learning approaches by using normalized negative log-likelihood to estimate uncertainty for unlabeled sentences and stratifying the data based on these scores. The final score is a weighted sum of the uncertainty score and the cosine distance between the sentence’s BERT embedding and its corresponding stratum centroid embedding. **HADAS** (Xia et al., 2024), introduced for Abstractive Text Summarization (ATS), assesses the susceptibility of a generative model to hallucinations across semantic frame, discourse, and content verifiability errors. It combines entailment-based semantic frame scoring, QA-based discourse evaluation, and BERTScore-based content verifiability assessment to produce a hallucination-aware score for each text. **LD-CAL** (Li et al., 2024), also designed for the ATS task, fuses curriculum learning and active learning by leveraging an LLM-determined difficulty score to partition documents into four levels and then selecting representative instances that maximize the model’s certainty gain, thus covering both high- and low-density regions.

Some works favor ED approaches since the query phase of AL in NLG can incur significant overhead, especially when the model is required to generate some text to assess the informativeness of the instance. **IDDs** (Tsvigun et al., 2022a) selects instances with low semantic similarity with the labeled pool and high similarity with the whole unannotated pool. **Bhatt et al. (2024b)** suggest using traditional submodular functions for subset selection. They demonstrate the effectiveness of the facility location function (Mirchandani and Francis, 1991) in some settings. We will refer to this strategy as “Facility Location” throughout the paper.

2.2 Existing AL Frameworks for NLP

There exist many libraries that allow running AL for various NLP tasks (Klie et al., 2018; Lin et al., 2019; Nguyen et al., 2022; Tsvigun et al., 2022b; Schröder et al., 2023). However, these systems miss many practical features and tools, crucial for seamless integration with data analysis pipelines and annotation. Features essential for seamless data annotation in text generation tasks with active

learning integration are predominantly scattered across various frameworks. **Huang (2021); Beck et al. (2021)** implement many state-of-the-art strategies for classification tasks. **Schröder et al. (2023)** provides unified interfaces for benchmarking AL on text classification datasets. **ALToolbox (Tsvigun et al., 2022b)** provides a pre-implemented set of AL strategies and a GUI for text annotation tasks such as text classification and information extraction. It also allows benchmarking AL strategies for encoder-based and sequence-to-sequence models. The tool proposed by **Golde et al. (2023); Human Signal (2023)** allows annotating data using LLMs, but has no integration with AL. Finally, **Argilla (Daniel and Francisco, 2023)** offers a comprehensive tool for data annotation and quality improvement in AI projects, but also lacks AL workflow support.

Additionally, running an AL loop with modern LLMs is both time- and memory-consuming and requires enabling approaches for efficient fine-tuning and inference. To our knowledge, **ATGen** is the first framework to synergize PEFT approaches like LoRA and inference-efficient frameworks like vLLM (Kwon et al., 2023) for efficient AL.

3 ATGen Description

In AL, one starts with a small labeled dataset and a large pool of unlabeled data. An acquisition model is trained on the labeled set, then used to evaluate the unlabeled data. A selection AL strategy is used to identify the most informative instances, which are then labeled by an oracle (a human or a model). This process repeats iteratively, gradually improving the model’s performance until some stopping criteria are met.

ATGen supports all stages of AL in NLG. It includes: (1) a web application for manual annotation with integrated AL support; (2) automatic AL-guided data annotation using LLMs, optimized for cost-efficient API usage; (3) a wide range of implemented AL query strategies, evaluation metrics, and configurable stopping criteria; (4) tools for efficient model fine-tuning and inference; (5) a user-friendly web interface for configuration and monitoring; and (6) benchmarking scripts for evaluating and comparing AL strategies across tasks and domains.

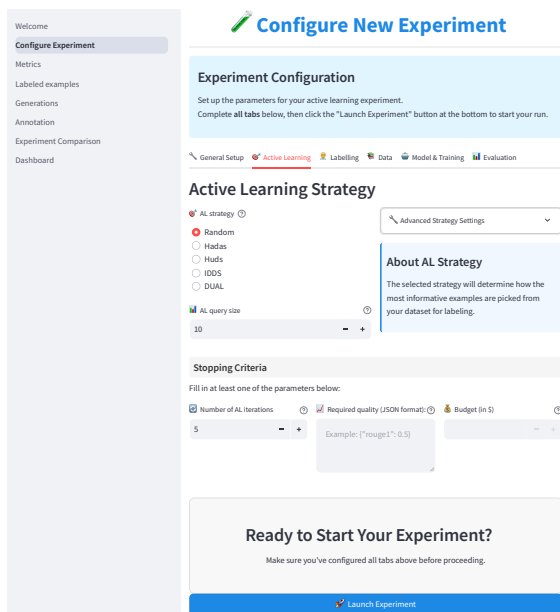


Figure 1: The ATGen configuration form to launch active learning in a GUI.

3.1 Framework Features

3.1.1 AL Strategies for NLG Tasks

ATGen implements all the state-of-the-art AL selection strategies in NLG (see Section 2.1). We also incorporate various uncertainty-based strategies, such as normalized sequence probability (Ueffing and Ney, 2007), mean token entropy (Zhao et al., 2020), and others.

3.1.2 Web GUI for Manual Labeling

ATGen can be used for manual text annotation via a web GUI application. For manual annotation, we recommend using ED strategies, as they require only a single round of annotation before training the target model. This approach significantly reduces annotation time and delays, making it especially suitable for scenarios involving human annotators. The GUI for annotation is displayed in Figure 2.

3.1.3 Automatic Labeling using LLMs

Users can select any chat-based model from a range of providers to serve as an annotator in place of a human. ATGen integrates seamlessly with leading API-based LLM providers, including **OpenAI**, **Anthropic**, and other OpenAI-compatible platforms such as Nebius AI Studio. For optimal annotation quality, we suggest using Claude-3.5-sonnet or GPT-4o models. For OpenAI-based models, we implement the batched API, which is 50% cheaper and several times faster compared to its synchronous analogue. Users can also choose

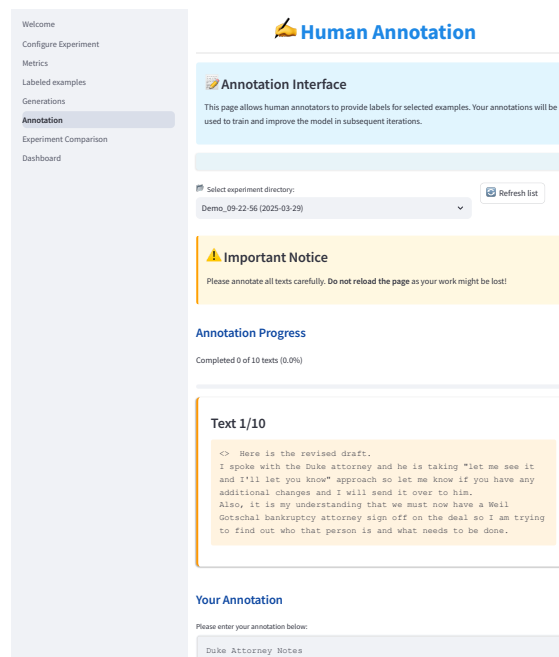


Figure 2: Human annotation page interface.

a model from HuggingFace or supply a custom model for data annotation. The selected model runs locally on the user’s hardware and processes the input data in batch mode.

3.1.4 Supported Acquisition Models and Datasets

The framework is tightly integrated with HuggingFace. It supports any acquisition model available through the HuggingFace Transformers library (Wolf et al., 2019) and allows pulling data from the HuggingFace hub via the datasets library (Lhoest et al., 2021). Users can also upload their own datasets in either CSV or JSON format.

3.1.5 Efficient Fine-Tuning and Inference

Most AL strategies require fine-tuning and/or inference with the target model. Since modern LLMs have billions of parameters, the implementation of computationally efficient methods for training and inference becomes crucial for real-world applications of the framework. ATGen, therefore, supports several parameter-efficient fine-tuning methods (Houlsby et al., 2019): LoRA (Hu et al., 2022), which approximates the update matrix as the product of two low-rank matrices; QLoRA (Dettmers et al., 2023), which further reduces memory usage using the 4-bit NormalFloat data type, double quantization, and paged optimizer; and DoRA (Liu et al., 2024), which allows for more expressive fine-tuning by introducing an additional low-rank matrix to model both additive and multiplicative

updates. The user can omit the usage of PEFT; yet, this will require a large amount of GPU memory.

For efficient inference, ATGen leverages three modern inference-accelerating frameworks: **vLLM**, which optimizes the inference through PagedAttention (Kwon et al., 2023), **SGLang** (Zheng et al., 2024), which leverages RadixAttention for prefix caching along with other techniques, and **Unsloth** (Daniel Han and team, 2023), which accelerates the inference through various optimizations like memory-efficient kernels.

3.1.6 Supported Evaluation Metrics

Performance evaluation in NLG tasks is a crucial bottleneck, since there are many perspectives from which the quality of the generation can be gauged (Yuan et al., 2021). We split the implemented metrics into three groups:

1. Automated metrics. These are traditional metrics such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and others, aimed at automatic evaluation of results.
2. Open-source LLM-based metrics. This group incorporates metrics that require the usage of an open-source model, such as BERTScore (Zhang* et al., 2020), BARTScore (Yuan et al., 2021), AlignScore (Zha et al., 2023), and others.
3. Proprietary LLM-based metrics. Recently, quality estimation with LLMs gained much attention and has been adopted in many works (Liu et al., 2023). We therefore use the DeepEval (Ip and Vongthongsri, 2025) framework for evaluation via LLMs. We note that this type of evaluation incurs additional computational cost compared to other methods. Therefore, we recommend using it only at the final stage of active learning for the ultimate assessment of model performance.

3.2 Demo Web Application

ATGen allows a user to deploy a web application for AL annotation on the user’s dataset with either human or LLM serving as an annotator in just one line of code. To launch AL annotation from the GUI, a user can configure a labeler, the data for annotation, and a stopping criterion from a web form (Figure 1). The application supports several stopping criteria: annotating a fixed number of texts, reaching a certain level of the model’s performance

on a test set, or running out of budget when using a human or an API-based LLM agent for annotation.

There are many other parameters when running AL: training hyperparameters, PEFT-related parameters, and others. To customize additional parameters, a user can create a configuration file and apply it using the submission form in the top left corner.

After each AL iteration, the model performance is evaluated either on the test data, if available, or on the test split of the labeled data.

3.3 Benchmark for AL Selection Strategies

ATGen provides benchmarking scripts for a seamless evaluation of the performance of AL strategies in NLG tasks. Running an experiment requires implementing the custom strategy according to the guidelines. The benchmarking tool can also be leveraged to test the existing approaches in various AL settings (e.g. in new domains, with LLM annotators, etc.). The example code to benchmark a strategy is provided in Figure 7 in Appendix C.

4 Experiments

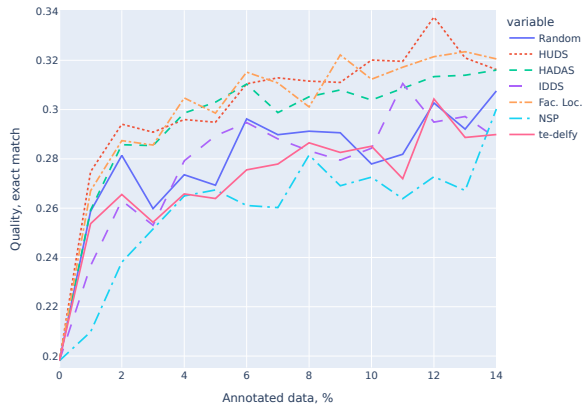
Using the ATGen benchmark, we evaluate the performance of AL and ED methods in various setups.

4.1 Experimental Setup

4.1.1 AL Settings

We adopt the widely used simulated AL experimental setup (Settles and Craven, 2008; Shen et al., 2017; Tsvigun et al., 2022a), which emulates the AL annotation cycle. At each iteration, we select a fixed number of top-ranked instances from the unlabeled pool according to the AL query strategy and assign them their ground-truth outputs, simulating an annotation by an oracle. The selected instances are removed from the unlabeled pool and added to the training dataset for subsequent iterations. We then train a new acquisition model from the previous checkpoint using the accumulated training data and evaluate its performance on the test set. This process produces a curve that illustrates how model performance depends on the amount of annotated training data. A higher curve indicates better performance of the AL query strategy, as it reflects the model’s ability to achieve better results with less training data. For robustness, we run each experiment several times with different random seeds and average the obtained curves.

Given the growing interest in LLMs application for data labeling (Honovich et al., 2023), we also



a) TriviaQA dataset with emulation of “manual” labeling.



b) TriviaQA dataset with labeling using an LLM.

Figure 3: Performance of AL selection strategies on the TriviaQA dataset with different annotation sources.

conduct experiments where an LLM annotates the data instead of using ground-truth annotations. In this scenario, we use DeepSeek-R1 as the annotation model.

4.1.2 Datasets, Metrics, and Acquisition Model

Following the recent works in this area (Tsvigun et al., 2022a; Xia et al., 2024), we evaluate AL and ED strategies on 4 diverse NLG tasks: open-domain question answering: TriviaQA (Joshi et al., 2017), Wiki subset; mathematical reasoning: GSM8K (Cobbe et al., 2021); reading comprehension: RACE (Lai et al., 2017); and text summarization: AESLC (Zhang and Tetreault, 2019). For TriviaQA and GSM8K, we select 1% of texts from the train set for labeling on each AL iteration. For RACE and AESLC, we select 10 texts to label on each AL iteration to align with the previous works.

Due to space limitations, we present results for the TriviaQA and GSM8K datasets in the main part of the paper, while results for RACE and AESLC are provided in Appendix A.

We perform experiments with an emulation of manual labeling on all datasets. On TriviaQA and GSM8K, we also conduct experiments with LLM-based labeling by DeepSeek-R1.

We run the experiments with the Qwen/Qwen3-1.7B acquisition model² – one of the state-of-the-art models to date in its size. The hyperparameters are presented in Appendix B.

To assess the performance of the acquisition model, we use the exact match (EM) metric for GSM8K and RACE, the relaxed version of EM that accepts any of the valid answers for TriviaQA, and

ROUGE-2 (Lin, 2004) along with AlignScore (Zha et al., 2023) for AESLC to ensure that the increased ROUGE score is not caused by an increase in hallucinations.

4.2 Results

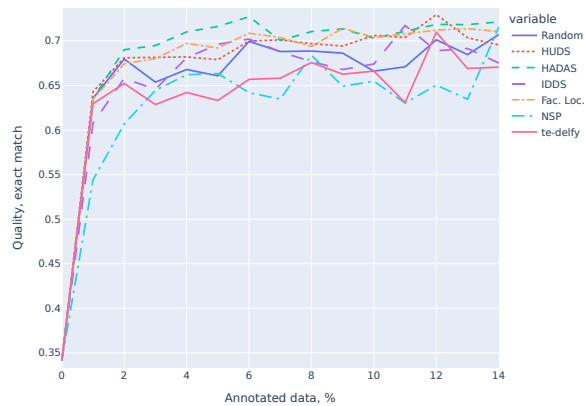
Figure 3 presents the performance of AL query strategies on the TriviaQA dataset under both manual annotation emulation and LLM-based annotation scenarios. The results reveal consistent patterns across both settings, with HUDS, HADAS, and Facility Location strategies substantially outperforming random sampling across all iterations. For example, random sampling requires over 12% of the dataset to be labeled to match the performance level that AL achieves with three times less data (just 4%) – in both the “manual” and LLM-based annotation scenarios.

Figure 4 shows analogous experiments on the GSM8K dataset. Under manual annotation emulation (Figure 4a), the same three strategies plus IDDS demonstrate superior performance compared to random sampling throughout all iterations. However, when using LLM-based annotation (Figure 4b), we observe a significant degradation in overall quality across all strategies. While HUDS, HADAS, and Facility Location maintain the advantage over random sampling, absolute quality scores decrease by several percentage points. The performance gap likely stems from the inherent limitations of the oracle, DeepSeek-R1. Despite being a state-of-the-art LLM on mathematical reasoning tasks, it is still prone to making annotation errors that accumulate through the AL process. This finding underscores that for specialized domains, human annotation remains crucial for obtaining

²<https://huggingface.co/Qwen/Qwen3-1.7B>



a) GSM8K dataset with emulation of “manual” labeling.



b) GSM8K dataset with labeling using an LLM.

Figure 4: Performance of AL selection strategies on the GSM8K dataset with different annotation oracles.

high-quality models.

Additional experiments on RACE and AESLC datasets (Figures 6 and 5 in Appendix A) corroborate these findings. HUDS, HADAS, and Facility Location consistently outperform random sampling across diverse NLG tasks. These strategies substantially reduce annotation costs by achieving target quality levels with far fewer labeled examples.

Overall, the results demonstrate that AL works effectively in both manual annotation and LLM-based annotation scenarios. This means that AL can reduce costs for LLM API calls by 2-4x when AL annotation is executed in a fully automatic regime, while achieving the same level of performance. Although AL requires retraining a small LLM on each iteration, which consumes computational resources, this process can be executed on a user’s hardware, making it effectively “free” for the user. Therefore, despite the additional computational expenses, the savings on LLM API calls are significantly more substantial.

5 Conclusion

We have presented ATGen – a comprehensive framework for AL in text generation tasks. The framework implements all state-of-the-art active learning techniques for NLG, offers a web-based annotation tool that supports both human and LLM-assisted labeling, and includes scripts for consistent experimental evaluation of AL query strategies. We have also conducted experiments, demonstrating that state-of-the-art strategies like HUDS can save annotators’ time and budget for LLM API calls.

We believe that AL is a promising approach even in the era of powerful LLMs, as it can help to reduce costs for building smaller models that

could be deployed in production. We hope that our framework will foster the development of better AL strategies in the future.

Limitations

We have not investigated possible bias introduced by active learning during annotation. This is an important future work as AL strategies might alter the data distribution significantly.

We note that AL requires some additional computational expenses for re-training the target LLM. If the target LLM is not big, these expenses might be negligible. However, for bigger LLMs, that might be an additional concern.

Ethical Considerations

For experiments and demo implementation, we reused pre-existing corpora and LLMs, which have been publicly released and approved for research purposes. The code of the demo has been released under the MIT license on [GitHub](#).

Using LLMs for automatic annotation should be approached with caution, as these models inherit social biases and often produce hallucinations. Hence, additional verification of annotation quality is required.

Acknowledgements

The work of A. Tsvigun, T. Bekleutov, R. Grigorev, R. Kuleev, and I. Makarov on Sections 2–4 was supported by the Research Center of the Artificial Intelligence Institute at Innopolis University and financially by the Ministry of Economic Development of the RF (code 25-139-66879-1-0003).

References

- Anthropic. 2025. [Introducing Claude 4](#).
- 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.
- Nathan Beck, Suraj Kothawade, Durga Sivasubramanian, Apurva Dani, Rishabh Iyer, and Ganesh Ramakrishnan. 2021. [Distil: Deep diversified interactive learning](#). <https://github.com/decile-team/distil>.
- Gantavya Bhatt, Yifang Chen, Arnav Mohanty Das, Jifan Zhang, Sang T. Truong, Stephen Mussmann, Yinglun Zhu, Jeffrey A. Bilmes, Simon S. Du, Kevin G. Jamieson, Jordan T. Ash, and Robert D. Nowak. 2024a. [An experimental design framework for label-efficient supervised finetuning of large language models](#). *CoRR*, abs/2401.06692.
- Gantavya Bhatt, Yifang Chen, Arnav Mohanty Das, Jifan Zhang, Sang T. Truong, Stephen Mussmann, Yinglun Zhu, Jeffrey A. Bilmes, Simon S. Du, Kevin G. Jamieson, Jordan T. Ash, and Robert D. Nowak. 2024b. [An experimental design framework for label-efficient supervised finetuning of large language models](#). *CoRR*, abs/2401.06692.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. [Training verifiers to solve math word problems](#). *arXiv preprint arXiv:2110.14168*.
- Vila-Suero Daniel and Aranda Francisco. 2023. [Argilla - Open-source framework for data-centric NLP](#).
- Michael Han Daniel Han and Unsloth team. 2023. [Unsloth](#).
- DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanbiao Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yudian Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *Preprint*, arXiv:2501.12948.
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. [Qlora: Efficient finetuning of quantized llms](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Jonas Golde, Patrick Haller, Felix Hamborg, Julian Risch, and Alan AKBik. 2023. [Fabricator: An open source toolkit for generating labeled training data with teacher llms](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023 - System Demonstrations, Singapore, December 6-10, 2023*, pages 1–11. Association for Computational Linguistics.
- Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023. [Unnatural instructions: Tuning language models with \(almost\) no human labor](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14409–14428, Toronto, Canada. Association for Computational Linguistics.
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. [Parameter-efficient transfer learning for nlp](#). In *In-*

- ternational conference on machine learning, pages 2790–2799. PMLR.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. **Lora: Low-rank adaptation of large language models**. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.
- Junjie Hu and Graham Neubig. 2021. **Phrase-level active learning for neural machine translation**. In *Proceedings of the Sixth Conference on Machine Translation*, pages 1087–1099, Online. Association for Computational Linguistics.
- Kuan-Hao Huang. 2021. Deepal: Deep active learning in python. *arXiv preprint arXiv:2111.15258*.
- Human Signal. 2023. Adala: A framework for autonomous data labeling agents. <https://github.com/HumanSignal/Adala>.
- Jeffrey Ip and Kritin Vongthongsri. 2025. **deepeval**.
- Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. **TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension**. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.
- Jan-Christoph Klie, Michael Bugert, Beto Boulosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. **The INCEpTION platform: Machine-assisted and knowledge-oriented interactive annotation**. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9, Santa Fe, New Mexico.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. **Efficient memory management for large language model serving with pagedattention**. In *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP 2023, Koblenz, Germany, October 23-26, 2023*, pages 611–626. ACM.
- Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. **RACE: Large-scale Reading comprehension dataset from examinations**. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.
- David D. Lewis and William A. Gale. 1994. **A sequential algorithm for training text classifiers**. In *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Dublin, Ireland, 3-6 July 1994 (Special Issue of the SIGIR Forum)*, pages 3–12. ACM/Springer.
- Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gungjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. **Datasets: A community library for natural language processing**. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. 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, Miami, Florida, USA, November 12-16, 2024*, pages 8959–8971. Association for Computational Linguistics.
- Bill Yuchen Lin, Dongho Lee, Frank F. Xu, Ouyu Lan, and Xiang Ren. 2019. **Alpacatag: An active learning-based crowd annotation framework for sequence tagging**. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL), Demo Track*.
- 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.
- Qiang Liu, Yanqiao Zhu, Zhaocheng Liu, Yufeng Zhang, and Shu Wu. 2021. **Deep active learning for text classification with diverse interpretations**. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pages 3263–3267. ACM.
- Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. 2024. **Dora: Weight-decomposed low-rank adaptation**. *CoRR*, abs/2402.09353.
- Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. **G-eval: NLG evaluation using gpt-4 with better human alignment**. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. 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, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 650–663. Association for Computational Linguistics.
- Pitu B. Mirchandani and Richard L. Francis, editors. 1991. *Discrete Location Theory*. Wiley.
- Birger Moëll. 2024. Comparing the efficacy of GPT-4 and chat-gpt in mental health care: A blind assessment of large language models for psychological support. *CoRR*, abs/2405.09300.
- Minh Van Nguyen, Nghia Ngo, Bonan Min, and Thien Nguyen. 2022. **FAMIE: A fast active learning framework for multilingual information extraction**. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: System Demonstrations*, pages 131–139, Hybrid: Seattle, Washington + Online. Association for Computational Linguistics.
- OpenAI. **Introducing o3 and o4-mini**.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. **Bleu: a method for automatic evaluation of machine translation**. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.
- Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. **Instruction tuning with gpt-4**. *Preprint*, arXiv:2304.03277.
- 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, EMNLP 2023, Singapore, December 6-10, 2023*, pages 9862–9877. Association for Computational Linguistics.
- Nicholas Roy and Andrew McCallum. 2001. Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 441–448. Morgan Kaufmann.
- Christopher Schröder, Lydia Müller, Andreas Niekler, and Martin Potthast. 2023. **Small-text: Active learning for text classification in python**. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 84–95, Dubrovnik, Croatia. Association for Computational Linguistics.
- Christopher Schröder, Andreas Niekler, and Martin Potthast. 2022. **Revisiting uncertainty-based query strategies for active learning with transformers**. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2194–2203, Dublin, Ireland. Association for Computational Linguistics.
- Ozan Sener and Silvio Savarese. 2018. **Active learning for convolutional neural networks: A core-set approach**. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.
- Burr Settles. 2009. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.
- Burr Settles and Mark Craven. 2008. **An analysis of active learning strategies for sequence labeling tasks**. In *2008 Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, Proceedings of the Conference, 25-27 October 2008, Honolulu, Hawaii, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1070–1079. Association for Natural Language Processing.
- Artem Shelmanov, Dmitry 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, EACL 2021, Online, April 19 - 23, 2021*, pages 1698–1712. Association for Computational Linguistics.
- Yanyao Shen, Hyokun Yun, Zachary Lipton, Yakov Kromrod, 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.
- 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. 2022a. **Active learning for abstractive text summarization**. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 5128–5152. Association for Computational Linguistics.
- Akim Tsvigun, Leonid Sanochkin, Daniil Larionov, Gleb Kuzmin, Artem Vazhentsev, Ivan Lazichny, Nikita Khromov, Danil Kireev, Aleksandr Rubashevskii, Olga Shahmatova, Dmitry V. Dylov, Igor Galitskiy, and Artem Shelmanov. 2022b. **ALToolbox: A set of tools for active learning annotation of natural language texts**. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 406–434, Abu Dhabi, UAE. Association for Computational Linguistics.
- Akim Tsvigun, Artem Shelmanov, Gleb Kuzmin, Leonid Sanochkin, Daniil Larionov, Gleb Gusev, Manvel Avetisian, and Leonid Zhukov. 2022c. **Towards computationally feasible deep active learning**.

- In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1198–1218, Seattle, United States. Association for Computational Linguistics.
- Nicola Ueffing and Hermann Ney. 2007. [Word-level confidence estimation for machine translation](#). *Comput. Linguistics*, 33(1):9–40.
- Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. [Self-instruct: Aligning language models with self-generated instructions](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. [Huggingface’s transformers: State-of-the-art natural language processing](#). *CoRR*, abs/1910.03771.
- Yu Xia, Xu Liu, Tong Yu, Sungchul Kim, Ryan A. Rossi, Anup B. Rao, Tung Mai, and Shuai Li. 2024. [Hallucination diversity-aware active learning for text summarization](#). *CoRR*, abs/2404.01588.
- Tim Z. Xiao, Aidan N. Gomez, and Yarin Gal. 2020. [Wat zeï je? detecting out-of-distribution translations with variational transformers](#). *CoRR*, abs/2006.08344.
- Michelle Yuan, Hsuan-Tien Lin, and Jordan Boyd-Graber. 2020. [Cold-start active learning through self-supervised language modeling](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7935–7948, Online. Association for Computational Linguistics.
- Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. [BartScore: Evaluating generated text as text generation](#). In *Advances in Neural Information Processing Systems*, volume 34, pages 27263–27277. Curran Associates, Inc.
- Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. [AlignScore: Evaluating factual consistency with a unified alignment function](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11328–11348, Toronto, Canada. Association for Computational Linguistics.
- Rui Zhang and Joel R. Tetreault. 2019. [This email could save your life: Introducing the task of email subject line generation](#). In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 446–456. Association for Computational Linguistics.
- Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. [BertScore: Evaluating text generation with bert](#). In *International Conference on Learning Representations*.
- Yuekai Zhao, Haoran Zhang, Shuchang Zhou, and Zhihua Zhang. 2020. [Active learning approaches to enhancing neural machine translation: An empirical study](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 1796–1806. Association for Computational Linguistics.
- Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark W. Barrett, and Ying Sheng. 2024. [Sglang: Efficient execution of structured language model programs](#). In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

A Additional Experiments

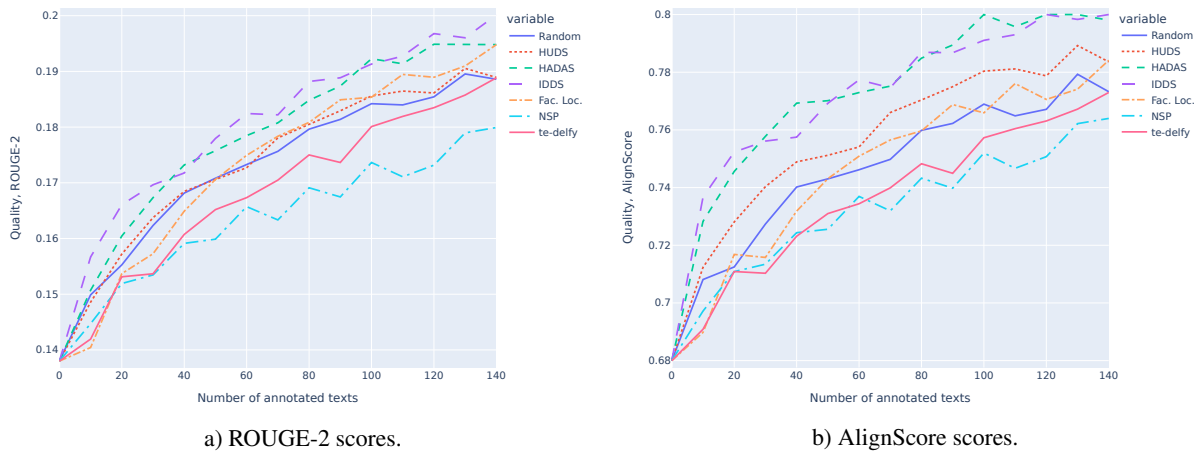


Figure 5: Performance of AL and ED strategies with emulation of “manual” labeling on AESLC in terms of the main metric (ROUGE-2) and a hallucination-sensitive metric (AlignScore).



Figure 6: Performance of AL and ED strategies with emulation of “manual” labeling on the RACE dataset.

B Model Hyperparameters for Benchmarks

Hparam	Value
Checkpoint	Qwen/Qwen3-1.7B
# Param.	1.7B
Quantization	None
Number of epochs	5
Train Batch size	16
Evaluation Batch size	16
Evaluation Split Size	20%
Gradient Accumulation Steps	1
Learning Rate	3e-5
Warmup Ratio	0.03
Weight Decay	0.01
Max. Gradient Norm	1
Early Stopping Patience	5
Optimizer	adamw_hf
Inference Framework	vLLM
Batch Size	16
GPU Memory Utilization	0.5
Temperature	0
Generation top_p	0.5
PEFT	Enabled
r	16
lora_alpha	16
lora_dropout	0.
LoRA bias	'none'

Table 1: Hyperparameter values and checkpoints from the Hugging Face repository (Wolf et al., 2019) of the models.

C Code Examples

```
HYDRA_CONFIG_NAME=base python scripts/run_active_learning.py al=STRATEGY_NAME
```

Figure 7: A Bash command example to benchmark a AL strategy with the name “STRATEGY_NAME”.

```
HYDRA_CONFIG_NAME=base run-al \  
  data=gsm8k \  
  al.init_query_size=0.01 \  
  al.query_size=0.01 \  
  al.num_iterations=20 \  
  al=huds \  
  evaluation.additional_metrics=[] \  
  labeller=api_llm \  
  labeller.parameters.model=gpt-4.1 \  
  labeller.parameters.max_tokens=4096 \  
  al.budget=100 \  
  labeller.price.input_per_1m=2 \  
  labeller.price.output_per_1m=8 \  
  labeller.api_key=<your_openai_api_key>
```

Figure 8: Advanced Bash code example to benchmark the strategy “huds” on the dataset “TriviaQA”, annotating 1% of texts on each iteration, with GPT-4.1 LLM serving as labeller, calculating only the ‘relaxed’ exact match metric.